9-13-2007

A General Quantum Mechanical Method to Predict Positron Spectroscopy

Paul E. Adamson

A GENERAL QUANTUM MECHANICAL METHOD TO PREDICT POSITRON SPECTROSCOPY

DISSERTATION

Paul E. Adamson, Captain, USAF

AFIT/DS/ENP/07-04

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/DS/ENP/07-04

A GENERAL QUANTUM MECHANICAL METHOD TO PREDICT POSITRON SPECTROSCOPY

DISSERTATION

Presented to the Faculty

Department of Engineering Physics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Paul E. Adamson, B.S.Chem.E., M.S.Env.E.

Captain, USAF

June 2007

AFIT/DS/ENP/07-04

A GENERAL QUANTUM MECHANICAL METHOD TO PREDICT POSITRON SPECTROSCOPY

Paul E. Adamson, B.S.Chem.E., M.S.Env.E.
Captain, USAF

Approved:

Larry W. Burggraf (Chairman)	Date
Milton E. Franke (Dean's Representative)	Date
James C. Petrosky (Member)	Date
Xiaofeng F. Duan (Member)	Date
Michael A. Marciniak (Member)	Date
Mark E. Oxley (Member)	Date

Accepted:

M. U. Thomas Dean, Graduate School of Engineering and Management	Date

AFIT/DS/ENP/07-04

Abstract

The nuclear-electronic orbital (NEO) method was modified and extended to positron systems for studying mixed positronic-electronic wavefunctions. These methods include: Hartree-Fock (HF); second-order Möller-Plesset perturbation theory (MP2); configuration interaction (CI); complete active space self-consistent field (CASSCF); and full configuration interaction (FCI). The methodology for calculating positron-electron annihilation rates based on NEO-HF and NEO-MP2 wavefunctions was also implemented. Positronic and electronic basis sets were optimized at the NEO-FCI level for the the positronium hydride (PsH) system and used to compute NEO-MP2 energies and annihilation rates. The effects of basis set size on correlation energies captured with the NEO-MP2 and NEO-FCI methods are compared and discussed.

Equilibrium geometries and vibrational energy levels were computed for the LiX and e^+LiX (X = H, F, Cl) systems at the MP2 and NEO-MP2 levels. Anharmonic effects were included by fitting the computed potential energy curves (PECs) to a Morse potential function. It was found that anharmonicity plays a significant role, specifically in the differences between the vibrational energy levels of the LiX and e^+LiX systems. The implications of these results with respect to vibrational Feshbach resonances (VFRs) for these systems is discussed.

The positron lifetime in pressed samples of $K_2B_{12}H_{12}{\cdot}CH_3OH$ was measured to be 0.2645 ± 0.0077 ns. This result is interpreted with quantum mechanical calculations of $B_{12}H_{12}^{2-}$ and $e^+B_{12}H_{12}^{2-}$. Calculations reveal a spherically symmetric positronic wavefunction, with a peak in the positron density at the outside edge of the hydrogen atom cage. The experimentally determined annihilation rate corresponds to an effective number of electrons of 1.88, or 0.94 of the two electrons present in the $B_{12}H_{12}^{2-}$ dianion, indicating that there is significant positron density both inside and outside of the $B_{12}H_{12}^{2-}$ cage.

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol		Page
PsH	positronium hydride	iv
LiX	lithium halides (X = H, F, Cl)	iv
e^+LiX	positronic lithium halides (X = H, F, Cl)	iv
e^+	positron	iv
ε_k	positron or electron kinetic energy	4
ε_b	positron binding energy to molecule in VFR equations	4
β^+	beta-plus particle (or positron)	7
ν	neutrino, in sodium-22 decay scheme	7
Q-value	decay energy for a nuclear process	7
L_+	positron diffusion length	8
Ps	positronium	8
para-Ps	singlet (antiparallel spins) positronium	9
ortho-Ps	triplet (parallel spin) positronium	9
τ	positron lifetime	10
λ	annihilation rate	12
NaX	sodium halides (X = Cl, F, Br, I)	12
m_0	electron (or positron) rest mass	13
c	speed of light	13
$\mathbf{p}$	electron-positron pair momentum	13
E_T	sum of annihilation photon energies	13
h	Planck's constant	13
ν	photon frequency	13
E_b^-	electron binding energy	13
E_b^+	positron binding energy	13
ΔE	difference in annihilation photon energies	13

Symbol		Page
$p_{\parallel}$	parallel component of electron-positron momentum	14
$p_{\perp}$	perpendicular component of electron-positron momentum	14
θ	deviation from collinearity of annihilation photons	14
υ	vibrational energy level	14
J	rotational energy level	14
e^-	electron	14
$\Phi_{k+,k-}(\mathbf{p})$	two-photon momentum density	16
$\mathbf{k}_-$	electron wave number	16
$\mathbf{k}_+$	positron wave number	16
$\psi_{k\pm}$	wavefunctions of the electron-positron pair	16
$\mathbf{r}$	spatial coordinate	16
H'	annihilation operator	17
$a(\mathbf{p}_1)$ and $b(\mathbf{p}_2)$	electron and positron annihilation operators	17
δ	delta function	17
$\psi_-(\mathbf{r}_1)$	electron position-space field operator	17
$\psi_+(\mathbf{r}_2)$	positron position-space field operator	17
V	quantization volume	18
ϕ^e	electron Slater determinant	18
ϕ^p	positron Slater determinant	18
$\rho^{2\gamma}(\mathbf{p})$	two-photon momentum density (final form)	18
$n_{-(+)}$	electron (positron) density	19
q	electron charge	19
$\gamma[n_-(\mathbf{r})]$	enhancement factor	22
N_e	number of electrons	25
N_p	number of quantum nuclei (or positrons)	25
N_c	number of classical nuclei	25
H_{tot}	total Hamiltonian operator	25
Z	charge of classical nuclei (or positron)	25

Symbol		Page
i and j	electron indices	26
i' and j'	quantum nuclei (or positron) indices	26
A and B	classical nuclei indices	26
M	mass of quantum nuclei (or positron)	26
$h^e(i)$	electron one-particle operator	26
$h^p(i')$	electron one-particle operator	26
Ψ_{tot}	nuclear-electronic (or positronic-electronic) wavefunction	26
Φ_0^e	electron Slater determinant of spin orbitals	26
Φ_0^p	nuclear (or positron) Slater determinant of spin orbitals	26
χ_i^e	electron spin orbital	27
$\chi_{i'}^p$	nuclear (or positron) spin orbital	27
ψ_i^e	electron spatial orbital	28
$\psi_{i'}^p$	positron spatial orbital	28
$\hat{H}$	reference Hamiltonian	31
ζ	exponent of Gaussian primitive	33
α_l and β_l^k	even-tempered parameters of kth exponent of type l	33
N	number of doubly occupied electronic MOs	34
r_0	classical electron radius	35
$\phi_{i(i')}$	$i(i')$th electron (positron) MO	35
$S_{iji'j'}$	four-center integral	35
a	occupied electron MO index	35
r	virtual electron MO index	35
r'	virtual positron MO index	35
$\langle ii' \vert jj' \rangle$	electron-positron Coulomb integral	35
$\varepsilon_{i(i')}$	ith (i'th) electron (positron) MO eigenvalue	35
R_e	equilibrium bond length	47
R_0	bond length including zero point energy correction	48
R_{exp}	experimental bond length	51

Symbol		Page
Δ	dissociation energy	51
ω_e	harmonic constant	51
$\omega_e\chi_e$	anharmonic constant	51
R_{BB}	boron-boron bond length in $B_{12}H_{12}^{2-}$	55
R_{BH}	boron-hydrogen bond length in $B_{12}H_{12}^{2-}$	55
CH_3X	methyl halides (X = H, F, Cl)	71

List of Acronyms

ACAR	angular correlation of annihilation radiation
ASA	atomic sphere approximation
CASSCF	complete active space self-consistent field
CDB	coincidence Doppler broadening
CI	configuration interaction
DBAR	Doppler broadening of annihilation radiation
DMC	diffusion Monte Carlo
FCI	full configuration interaction
FWHM	full width at half maximum
GAMESS	General Atomic and Molecular Electronic Structure System
HF	Hartree-Fock
IMOMM	integrated molecular orbital/molecular mechanics
IP	ionization potential
IPM	independent particle model
LDA	local density approximation
LLNL	Lawrence Livermore National Laboratory
LMTO	linear-muffin-tin orbital
MCSCF	multiconfiguration self-consistent field
MESP	molecular electrostatic potential
MM	molecular mechanics
MO	molecular orbital
MP2	second-order Möller-Plesset perturbation theory
MRPT	multi-reference perturbation theory
NEO	nuclear-electronic orbital
NMR	nuclear magnetic resonance
PALS	positron annihilation lifetime spectroscopy

PAS	positron annihilation spectroscopy
PEC	potential energy curve
PES	potential energy surface
PMT	photomultiplier tube
QM	quantum mechanics
QMC	quantum Monte Carlo
RHF	restricted Hartree-Fock
ROHF	restricted open-shell Hartree Fock
SCF	self-consistent field
SIMOMM	surface integrated molecular orbital/molecular mechanics
SVM	stochastic variational method
TPMD	two-photon momentum density
UHF	unrestricted Hartree-Fock
VFR	vibrational Feshbach resonance
ZPDL	zero positron density limit

A GENERAL QUANTUM MECHANICAL METHOD TO PREDICT POSITRON SPECTROSCOPY

I. Introduction

Dirac [14] first predicted the existence of the positron as the antiparticle of the electron in 1928. In 1932, the first experimental indications of an unknown particle were found in cloud-chamber photographs of cosmic rays, and this particle was later identified as the positron. Annihilation of the positron with electrons in matter was first studied in the 1940s. An important early discovery was that energy and momentum conservation during the annihilation process could be utilized to study properties of solids. Currently, positron annihilation spectroscopy (PAS) experiments are widely used to study materials defects, including point defects in semiconductors and voids in composite materials. [31]

The Air Force is interested in PAS for a variety of reasons. For example, PAS has been used to measure damage levels in aerospace materials. Used in conjunction with uniaxial tensile strain testing and strain controlled fatigue testing, remaining life assessments were made for components such as turbine engine disks and blades, fuel injector stems inside auxiliary power unit gas generators, and structural components. [68] There is also interest in evaluating PAS as a tool for probing nanoscopic voids and defect structures of energetic materials in order to quantify their concentration and morphology.

Several sophisticated computational methods, such as the stochastic variational method (SVM) and the quantum Monte Carlo (QMC) method, have been employed to accurately compute positron and positronium binding to atoms. [52] As in conventional electronic structure calculations, however, more approximate methods for computing electron-positron wavefunctions must be employed as the size of the system becomes larger. In 1970, Schrader [51] presented a self-consistent field (SCF) theory for a single positron bound to many-electron systems that included electron-positron correlation explicitly in the electronic molecular orbitals (MOs). Kurtz and Jordan [33] studied positron-molecule com-

plexes with HF theory in 1981. Since then, several groups have employed various methods to study small molecular systems, and the most well-studied system is positronic lithium hydride, e^+LiH. [8, 39, 58] In order to study positronic atomic and molecular systems, a general method to incorporate positrons into MO calculations would be useful to advance the field of positron chemistry.

The objective of this dissertation is to introduce and illustrate a modified NEO method for calculations of positronic-electronic systems. The NEO approach was developed to include nuclear quantum effects directly within electronic structure calculations and has been successfully applied to proton transfer and hydrogen tunneling problems. [24,43,44,46,59–61, 71] Within the framework of the extension of the NEO method to positronic systems, a single positron is treated quantum mechanically in the General Atomic and Molecular Electronic Structure System (GAMESS) program. [49] At the NEO-HF level of theory, the positron is treated in the same way as are the electrons, and mixed positronic-electronic wavefunctions are calculated variationally using MO techniques. Electron-electron and electron-positron correlation can be calculated by MP2 and CASSCF theory. Also, routines for calculating electron-positron annihilation rates based on NEO-HF and NEO-MP2 have also been implemented in GAMESS. The FCI method is also available within the CASSCF framework by including all of the electronic and positronic MOs within the active space of the calculation.

Chapter II contains some background information on PAS, including an overview of positron transport and annihilation in condensed matter. In addition to traditional PAS techniques, experiments involving VFRs are also discussed. In Section 2.2, methods for modeling positron interaction with matter are presented, culminating in a description of the NEO method and the modifications required to handle positrons.

In Chapter III, the NEO method is applied to the PsH and e^+LiH systems. PsH consists of a classical proton, two quantum electrons, and a quantum positron. In the e^+LiH system, the lithium nucleus and the proton are treated classically, and the four electrons and the positron are treated quantum mechanically. For the PsH system, positronic and

electronic basis sets are first optimized within the NEO framework at the FCI level. These basis sets are then used to compute NEO-MP2 energies for PsH. The basis set dependence of the total energies and the electron-electron and electron-positron correlation energies at the NEO-MP2 and NEO-FCI levels are also assessed. For e^+LiH, even-tempered electronic and positronic basis sets for PsH and Li are first optimized at the NEO-MP2 and MP2 levels, respectively. At the same level of theory with a reoptimized positronic basis set, the PEC for e^+LiH is calculated and the equilibrium geometry and the shift of the vibrational frequency caused by addition of a positron to LiH is determined.

In Chapter IV, the vibrational energy levels of some polar diatomics are calculated with the modified NEO method. Calculations of vibrational energies for positronic systems are useful for explaining the enhanced positron annihilation rates observed for some molecular gases. This enhancement has been attributed to the existence of VFRs.

The term "Feshbach resonance" comes from the seminal paper of Herman Feshbach [18] which introduces a method for computing resonant cross sections. Such a resonance will occur in electron scattering experiments if the combined energy of the incident particle and the target molecule is degenerate (or nearly degenerate) with the energy of a combined state of the particle and the target molecule. The target molecule can be in any rovibronic[1] state before combining with the particle. Also, it is possible that the state of the combined particle/target system is an excited rovibronic energy level as well. In essence, the Feshbach resonance is an enhanced coupling to the combined state, and it often results in an increased cross section for an escape channel. For example, the cross section for dissociative attachment is enhanced if the energy of the combined state exceeds the dissociation energy.

Positron annihilation rates in molecular gases were first measured in the 1950s and 1960s, [12, 13, 53] and the results for oxygen, helium, and nitrogen were close to what was expected based on annihilation with a free electron gas of the same electron density. Surprisingly, the results for methane were a factor of two or three higher than the rate

[1] rotational, vibrational, and/or electronic

expected for a free electron gas. Paul and Saint-Pierre measured annihilation rates for hydrocarbons that were 20 to 200 times larger than the free electron rate, with the rate increasing rapidly with molecular size. They and others proposed the formation of positron-molecular ions to explain these findings. [19, 45, 55]

Since then, several different theories have been proposed to explain the enhanced annihilation rates. Recent advances by Barnes *et al.* [2] in the energy resolution of slow positron beams to 25 meV full width at half maximum (FWHM) showed that positron-electron annihilation rates of molecules are strongly enhanced at energies near their fundamental vibrational energies. Their measurements are strong evidence that the enhancements are due to the existence of VFRs.

A model for describing the enhanced annihilation rates due to VFR developed by Gribakin [22] treats the positron binding energy as a free parameter that can be fit to experimental measurements. In their model, the VFR enhancement of annihilation rates requires a bound state of the positron and the target molecule, and the resonance occurs when the energy of the unbound positron plus the ground state molecule is nearly degenerate with the vibrationally excited molecule and a bound positron:

$$E_{\upsilon=0} + \varepsilon_k = E_{\upsilon=1} - \varepsilon_b, \tag{1}$$

where $E_{\upsilon=0}$ and $E_{\upsilon=1}$ are the ground and vibrationally excited energies of the molecule, respectively, ε_k is the kinetic energy of the positron, and ε_b is the binding energy of the positron to the molecule. [2]

As described in Section 4.2, a more complete description of VFR enhancement of the annihilation rate requires knowledge of the vibrational energy levels of the positronic system, which in turn requires an accurate potential energy surface (PES). To date, PECs have been computed for only a handful of positronic diatomics, including e^+LiH, e^+BeO, and e^+LiCl. Studies of larger polyatomic systems will require considerable effort to develop highly parallelized codes that utilize techniques that have been developed for traditional electronic structure calculations. In Chapter IV, PECs for the e^+LiH, e^+LiF, and e^+LiCl

systems are calculated with the NEO-MP2 method modified to handle positrons. These curves are fit to Morse potential functions in order to determine the vibrational energy levels of the systems.

In Chapter V, the NEO method is applied to the relatively large system positronic dodecahydrododecaborate dianion, $e^{+}B_{12}H_{12}^{2-}$, to calculate the electronic and positronic wavefunctions. The results are used to interpret measurements of the positron lifetime in pressed pellets of potassium dodecahydrododecaborate methanolate, $K_2B_{12}H_{12}\cdot CH_3OH$. This large cage-like molecule was chosen because it will bind a positron in a well-defined quantum state. Also, the highly symmetric nature of the molecule makes the calculation of its electronic and positronic wavefunctions accessible.

In the icosahedral borane molecule, $B_{12}H_{12}$, each boron atom at the twelve vertices of an icosahedron is in six-fold coordination, with covalent bonds to an "external" hydrogen atom and five other "internal" boron atoms. This unusual bonding behavior for a group IIA element with a valence of three results in a charge accumulation about the center of the icosahedron's 20 triangular faces rather than along the lines that link adjacent boron atoms. Boron icosahedra readily form dianions and are the main constituents of a large host of insulating refractory solids. For example, in $B_{12}P_2$, $B_{12}As_2$, and $B_{12}O_2$, the B_{12} dianions are centered at the eight vertices of a rhombohedron's longest diagonal, the *c*-axis. Strong covalent bonds link six of the boron atoms of each icosahedron to neighboring icosahedra, and the remaining six boron atoms bond to the solid's cations. [16]

Icosahedral boron-rich solids are generally very hard and have high melting temperatures. These solids also survive extremely well in high radiation environments, and could find a variety of uses in aerospace applications. This resiliency has been attributed to a "self-healing" process in which a boron atom that is displaced from the icosahedron leaves behind an electron, resulting in a Coulomb attraction that facilitates recombination. The small size of the boron cation also aids in the recombination. [16] Also, the relative stability of the icosahedron relative to the distorted anion caused by knock-on displacement also contributes to the "self-healing". [69]

Appendix A contains the mathematical expressions for computing electron-positron annihilation observables from NEO-HF and NEO-MP2 wavefunctions. For completeness, expressions for NEO-HF and NEO-MP2 energies and wavefunctions are also presented. In Appendix B, a sample of the Mathematica® code that was used during the code development is given. The final modifications and additions to the GAMESS program for calculating positronic systems and annihilation rates are given in Appendix C.

II. Background

This dissertation attempts to connect quantum mechanical calculations of positronic systems to experimental observables in PAS, two very unique areas of research. For convenience, the following chapter contains a brief review of PAS and methods for modeling positron interactions with matter. In addition to the traditional forms of PAS, positron annihilation lifetime spectroscopy (PALS) and momentum distribution techniques, a discussion of VFRs is presented.

2.1 Positron Annihilation Spectroscopy

Radioactive decay of ^{22}Na (2.602 year half-life) through emission of a β^+ particle (or positron), illustrated in Figure 1, is the prominent source of positrons in the laboratory. The primary path for β^+ decay of ^{22}Na is written schematically as

$$^{22}_{11}\mathrm{Na} \rightarrow {}^{22}_{10}\mathrm{Ne}^* + \beta^+ + \nu \tag{2}$$

where ν is the neutrino. The neutrino has an extremely small interaction probability with matter, so it is undetectable for all practical purposes. The fixed decay energy, or Q-value, of 0.546 MeV for Equation (2) is shared between the β^+ particle and the "invisible" neutrino. The β^+ particle thus appears with an energy that varies from decay to decay and can range from zero to the "beta endpoint energy," which is numerically equivalent to the Q-value. Relaxation of the excited ^{22}Ne nucleus results in the emission of a 1.27 MeV "birth" γ-quantum. [28]

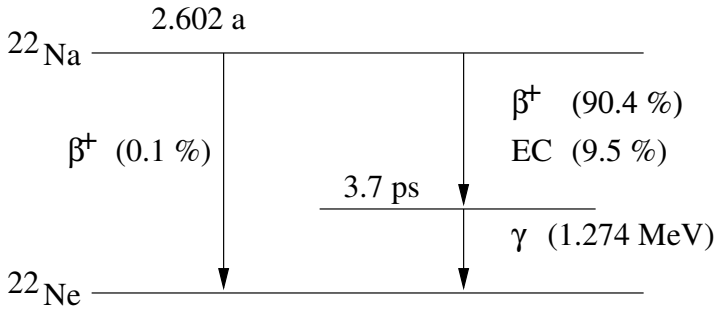

Figure 1: Decay scheme for ^{22}Na.

The ^{22}Na sources used for benchtop positron annihilation experiments are usually encapsulated in a thin metal disc or sealed between two pieces of mylar, and a "source-sample sandwich" configuration is normally used to maximize the number of positrons that annihilate in the sample. In this configuration, the two samples must be identically prepared, and for momentum distribution techniques (described in Section 2.1.2.2) their crystal lattice orientation must be matched. An alternative to the "source-sample sandwich" configuration, used to reduce the source contribution to the annihilation radiation, is deposition of the ^{22}Na solution directly onto the sample surface.

2.1.1 Positron Transport and Annihilation in Condensed Matter. Once a positron enters condensed matter, it thermalizes and annihilates with an electron with a lifetime on the order of a nanosecond. When the β^+ particle enters the material, its energy is much higher than thermal, and the energy transfer rate by core electron ionization is very high. Thus, the positron energy decreases rapidly through this thermalization process, which only lasts a few picoseconds. The thermalized positron diffuses through a crystal lattice, scattering from acoustic phonons, in much the same way as an electron diffuses in the absence of fields. The keV positron diffusion length, L_+, is about 100 nm.

In metallic solids and semiconductors, the thermalized positrons, at room temperature, are often trapped in negatively-charged defect sites such as vacancies, vacancy agglomerates, and dislocations. The wave function of the positron is localized in such a defect until it annihilates with an electron of the immediate surrounding. Since the electronic wave function where the annihilation occurs differs significantly from the bulk crystal electronic wavefunction, the annihilation radiation can be used to deduce information about the defect. [31]

2.1.1.1 Positronium Formation and Decay in Condensed Matter. Positronium (Ps), which represents a bound state between a positron and an electron, may be formed in or on the surface of condensed matter. The Schrödinger equation for Ps is identical to that for hydrogen, where the reduced mass of the hydrogen atom is replaced by one half of

the electron mass. As with hydrogen, Ps is formed in two states: the triplet (parallel spins) *ortho*-Ps and the singlet (antiparallel spins) *para*-Ps. The lifetimes of free *para*-Ps and *ortho*-Ps are 125 ps and 142 ns, respectively. [25]

The *para*-Ps annihilates intrinsically (*i.e.*, annihilation between the particles forming the Ps atom) mainly into two photons of 511 keV (termed two-photon annihilation); however, *ortho*-Ps annihilates in vacuum into three photons (three-photon annihilation) in order to conserve spin. In matter, the picture is quite different since the positron wave function overlaps with electrons outside of the Ps atom. The annihilation with such electrons having an antiparallel spin decreases the lifetime of *ortho*-Ps by about two orders of magnitude to about 1 to 5 ns. This process is called *pick-off* annihilation. [31]

Positronium formation is minimal in metallic and semiconductor materials, since the electron density must be extremely low for this to occur. In fact, delocalized Ps has only been found in very rigid solids, such as ice, and is usually found in a defect-free solid in a very localized, self-trapped state. This self-trapping occurs when the Ps atom pushes away the surrounding atoms, creating a cage for itself. In addition to self-trapping, Ps can also be captured in structural defects, such as vacancies and vacancy clusters. (Ps formation has been reported in porous silicon.) If no large open-volume defects are present, Ps may only exist at the semiconductor surface, and its occurrence may be taken as a measure of the fraction of positrons diffusing back to the surface. [31]

In molecular solids, Ps forms more readily, and the *ortho*-Ps lifetime can be used as a probe of the molecular structure of these systems. Before annihilation, the *ortho*-Ps will localize in a vacancy with a lifetime that is a function of the exchange interaction between the *ortho*-Ps electron and the electrons bound to the molecules surrounding the vacancy. The exchange probability decreases for larger vacancies, and the *ortho*-Ps lifetime increases.

2.1.1.2 Two-photon Annihilation in Condensed Matter. Practically speaking, the implantation energy of positrons from an encapsulated ^{22}Na source is so high that no positrons are diffusing back to the surface and no Ps can be formed. [31] For such a

case, one-photon annihilation can only occur near a heavy nucleus to absorb the recoil and thus has a small cross-section. Also, the cross-section for decay through emission of three or more photons is small and decreases rapidly as the number of photons increases. Hence, most positrons in condensed matter annihilate with the emission of two photons. [36] It is the detection of one or both of these annihilation photons that is the basis for PAS in materials.

2.1.2 Traditional Positron Annihilation Spectroscopy. In traditional PAS experiments, two types of information, lifetime and momentum, are available. In PALS, the lifetime of the positron in the sample of interest is determined. In Doppler broadening of annihilation radiation (DBAR) and angular correlation of annihilation radiation (ACAR) experiments, components of the momentum of the electron-positron pair that underwent annihilation are measured. These techniques are well-established and are powerful tools for the analysis of defects in condensed matter.

2.1.2.1 Positron Annihilation Lifetime Spectroscopy. In PALS, the time lapse between the detection of the 1.27 MeV birth gamma and one or both of the annihilation photons is taken to be the lifetime, τ, of the positron in the sample. The thermalization time for the positron is on the order of 1 ps, while the positron lifetime as a whole is on the order of 100 ps. Thus, the positron spends the vast majority of its life sampling the electronic structure of the crystal while at thermal energies (diffusing). In a defect-free metal crystal, the thermalized positron's wavefunction will be very delocalized, and it will overlap with the electronic wavefunction of the "bulk" crystal; however, as mentioned before, a thermalized positron may become localized in a negatively-charged trapping center such as a vacancy. Since the trapping rate is dependent on the concentration of such sites (the positron must encounter such a site within the $\sim$100 nm diffusion length), and the positron's lifetime in a trapping site is inversely proportional to the electron density at the site, the lifetime measurement can be used to characterize both the types of defects present and their concentration. The activity of the source must be sufficiently low in order to en-

sure that on average only one positron is in the sample at any time. This avoids intermixing of start and stop quanta originating from different annihilation events. [31]

In order to obtain the best time resolution, barium fluoride, BaF_2, or plastic scintillators and photomultipliers with a short pulse rise-time are used. The time resolution of the spectrometer is determined mainly by the scintillator-multiplier part and ranges between 180 and 280 ps. The practical consequence of this relatively poor resolution is that one is limited to the determination of positron lifetime components longer than about 50 ps; however, due to the low jitter of such systems, the positron lifetimes can be determined to an accuracy of about 1 ps.

The trapped positron states decay with exponential lifetimes, where the number of exponential decay terms is the number of types of defect sites plus one. The intensity of the longer lived component is proportional to the defect concentration. Positron lifetimes for many vacancy-type defects have been experimentally determined. By comparison with theoretical calculations, defect types can be identified by their specific positron lifetimes. This approach works best for specific types of defects, such as are generated by ion implantation in metals and semiconductors. Decomposition of the exponential decay into various lifetimes is difficult if several different types of defects make significant contributions to the spectrum, as in neutron-irradiated samples. [31]

A given system has only one annihilation rate, and the term "spin-averaged annihilation rate," that is sometimes seen in the literature, has no operational meaning and is not directly measurable. Regardless, it is a useful concept for comparing positron lifetime data from different systems, and Schrader [52] gives a description of this technique. If a positron is in an electron-rich environment in which all of the electron spins are paired (closed-shell), its wavefunction is described as *virtual* Ps, in which all of the electrons participate, one at a time. In such a configuration, the positron experiences electron spins parallel and antiparallel to its own equally. As described in Section 2.1.1, the parallel alignment annihilates by the slow three-photon process. The antiparallel alignment is equally likely to become quantized as either *ortho*-Ps ($S = 0$) or *para*-Ps ($S = 1$) at the moment of annihilation. Since

Table 1: Positron lifetimes (τ) and the corresponding effective numbers of electrons for water solutions of NaX (X = Cl, F, Br, I) as measured by Stoll *et al.* [57] The effective number of electrons for each system is computed by taking the ratio of the measured annihilation rate ($1/\tau$) to the ideal value of the spin-averaged annihilation rate from Equation (3).

X	τ (ps)	Effective Number of Electrons
F	1238	0.40
Cl	898	0.56
Br	662	0.75
I	625	0.80

the *para* annihilation process is much faster and dominates, the annihilation rate, λ, for the system is close to one-fourth that of isolated *para*-Ps. This is called the spin-averaged annihilation rate, and its ideal value is

$$\frac{1}{4}\lambda_{S=0} + \frac{3}{4}\lambda_{S=1} \approx 2.003\,\mathrm{ns}^{-1}. \tag{3}$$

While the above value is not the annihilation rate for any real systems, an estimate of the number of electrons gathered around the positron in closed-shell systems is its annihilation rate divided by the rate on the right side of Equation (3). This analysis has been caried out for PALS measurements of water solutions of NaX (X = Cl, F, Br, I) by Stoll *et al.* [57] For these systems, the annihilation rate increases in the expected order (NaI $>$ NaBr $>$ NaCl $>$ NaF). The lifetimes and corresponding numbers of effective electrons are given in Table 1.

2.1.2.2 Momentum Distribution Techniques. As stated earlier, the mass-energy transformation of an electron-positron pair in condensed matter mostly results in the emission of two photons. In the center-of-mass frame of reference of the electron-positron pair, the two photons are emitted in exactly opposite directions, and the energy of

each is exactly half of the energy released by the annihilation process:

$$E = \frac{1}{2} \cdot 2m_0c^2 = 511 \text{ keV}, \tag{4}$$

where m_0 is the electron (or positron) rest mass and c is the speed of light. [31] In the laboratory frame, the momentum of the electron-positron pair, $\mathbf{p}$, is transferred to the annihilation radiation, resulting in Doppler broadening and a deviation from collinearity of the annihilation photons. Either one or both of the annihilation photons are detected by high-purity germanium detectors in a DBAR experiment. The coincidence Doppler broadening (CDB) method, in which both photons are measured, results in a reduced background and overall better resolution. The sum of the annihilation photon energies, $E_T = h\nu_1 + h\nu_2$, is a Doppler-free measurement of the rest-mass of the electron-positron pair less the electron and positron binding energies:[1]

$$E_T = 2m_0c^2 - E_b^- - E_b^+ \tag{5}$$

$$\approx 2m_0c^2 \tag{6}$$

$$= 1022 \text{ keV}. \tag{7}$$

The difference in photon energies is equal to the momentum of the electron-positron pair in the direction of propagation of the annihilation photons:

$$\Delta E = h\nu_1 - h\nu_2 = \frac{\mathbf{p} \cdot \mathbf{c} - \frac{c^2\mathbf{p}^2}{E_T}}{1 - \frac{\mathbf{p} \cdot \mathbf{c}}{E_T}}, \tag{8}$$

where $\mathbf{c}$ is the velocity of light in the direction of $h\nu_1$. For thermal energies, Equation (8) reduces to $\Delta E = \mathbf{p} \cdot \mathbf{c}$. [23]

Since the deviation from collinearity is small, an approximation is made in the literature, and the Doppler broadening of the photons is related to the component of the mo-

[1] Weber *et al.* [72] omit the positron binding energy in their equation for E_T. Clearly, it is negligible, but it should be included prior to simplification.

mentum parallel to the photon propagation direction, $p_{\parallel}$, by the equation $\Delta E = p_{\parallel}c$. The component of the momentum perpendicular to the propagation direction, $p_{\perp}$, results in a deviation, θ, from collinearity of the annihilation photons. This deviation is measured in an ACAR experiment, and the relationship between the measurement and the electron-positron momentum is $\theta = p_{\perp}/m_0 c$. [31]

2.1.3 Vibrational Feshbach Resonances of Positrons with Molecular Gases. As mentioned in Chapter I, it has been known for some time that certain molecular gases exhibit unusually high positron annihilation rates, and recent advances in slow positron beam experiments by Barnes *et al.* have confirmed that the increase in the annihilation rate occurs when the positron that is interacting with the molecule has an energy in the range of the vibrational modes of the molecule. [2] A model to explain this phenomena, which has been developed by Gribakin, [22] requires the existence of VFR that are similar to those observed in electron scattering experiments.

Feshbach resonance enhancement to inelastic scattering, elastic scattering, and dissociative attachment (diss. att.) in electron scattering experiments for a molecule, AB, in a vibrational energy level, υ, and a rotational energy level, J, can be represented by a set of equations involving an intermediate state

$$\begin{aligned} \mathrm{AB}(\upsilon, J) + \mathrm{e}^-(\varepsilon_k) \rightarrow (\mathrm{AB}^-)^* &\xrightarrow{inelastic} \mathrm{AB}(\upsilon', J') + \mathrm{e}^-(\varepsilon_{k'}) \\ &\xrightarrow{elastic} \mathrm{AB}(\upsilon, J) + \mathrm{e}^-(\varepsilon_k) \quad . \\ &\xrightarrow{diss.\ att.} \mathrm{A} + \mathrm{B}^- \end{aligned} \tag{9}$$

If the coupling between the intermediate state, $(\mathrm{AB}^-)^*$, and the decay channels is strong, an increased population of this resonance state enhances the cross sections for the decay channels. [1]

As stated previously, VFR also explains the high positron annihilation rate of certain molecular gases. Of course, direct annihilation (dir. ann.) of the positron with an electron

in the target molecule is always available

$$\mathrm{AB}(\upsilon,J)+\mathrm{e}^{+}(\varepsilon_k)\xrightarrow{dir.\,ann.}\mathrm{AB}^{+}(\upsilon',J')+2\gamma \tag{10}$$

Also, an additional channel can be included with equation (9) to include resonant annihilation (res. ann.) of the positron with an electron in the target molecule

$$\begin{aligned}\mathrm{AB}(\upsilon,J)+\mathrm{e}^{+}(\varepsilon_k)\rightarrow(\mathrm{e}^{+}\mathrm{AB})^{*} &\xrightarrow{inelastic}\mathrm{AB}(\upsilon',J')+\mathrm{e}^{+}(\varepsilon_{k'})\\ &\xrightarrow{elastic}\mathrm{AB}(\upsilon,J)+\mathrm{e}^{+}(\varepsilon_k)\\ &\xrightarrow{res.\,ann.}\mathrm{AB}^{+}+2\gamma\end{aligned} \tag{11}$$

where $(\mathrm{e}^{+}\mathrm{AB})^{*}$ is the positronic molecule in an excited rovibronic state. If the electron-positron wavefunction overlap is enhanced compared to when the positron is in a free state, then the annihilation rate will be enhanced due to population of the intermediate state. [1]

There is also another channel leading to annihilation that is very unlikely. It involves dissociative attachment followed by annihilation, which will be called dissociative annihilation (diss. ann.)

$$\mathrm{AB}(\upsilon,J)+\mathrm{e}^{+}(\varepsilon_k)\rightarrow(\mathrm{e}^{+}\mathrm{AB})^{*}\xrightarrow{diss.\,att.}\mathrm{A}^{+}+\mathrm{PsB}\xrightarrow{diss.\,ann.}\mathrm{A}^{+}+\mathrm{B}+2\gamma \tag{12}$$

This channel is not mentioned in Ref. 1. As discussed in Section 2.2.4, if the ionization potential (IP) is large, $\mathrm{e}^{+}\mathrm{B}^{-}$ is formed in Equation (12) instead of PsB.

While scattering theory is necessary to describe the interactions of slow positrons with molecular gases in a rigorous way, stationary state potential energy surfaces can provide some very useful insights into VFR and the associated positron-molecule complexes. Such calculations would establish the stability of the system and the nature of the positron wavefunction. Barnes *et al.* [2] were extremely judicious in measuring VFR enhancement of the annihilation rate for several highly symmetric systems (*e.g.*, benzene and acetylene) and molecules with a modest number of particles (*e.g.*, ammonia and methane) in hopes that future modeling efforts could be directly compared to experiments.

2.2 Modeling Positron Interaction with Matter

Much progress has been made in the area of positron annihilation in matter since P.A.M. Dirac published his historic 1927 paper entitled "The Quantum Theory of the Emission and Absorption of Radiation," which opened the door to *quantum field theory*. [48] It was not until the 1940s that annihilation of the positron with electrons in matter was first studied; however, it was discovered early that energy and momentum conservation during the annihilation process could be utilized to study properties of solids. [31] Early experiments with positrons focused on probing the Fermi surface of metals and alloys. [3, 11] Eventually, it was discovered that positrons could be trapped in negatively charged defects such as vacancies, and insulator and semiconductor materials began to be investigated. The study of defects in compound and elemental semiconductors now amounts to about half of the total number of papers on defect studies with positrons. [31]

2.2.1 Quantum Field Theory Framework for Positron Annihilation Spectroscopy.

In 1950, DeBenedetti *et al.* [11] presented a method to calculate (to within a constant) the probability amplitude, $\Phi_{k+,k-}(\mathbf{p})$, for the emission of two photons with total momentum, $\mathbf{p}$, as a result of the annihilation of an electron and a positron with wave numbers, $\mathbf{k}_-$ and $\mathbf{k}_+$, respectively,

$$\Phi_{k+,k-}(\mathbf{p}) = \int_{\mathbb{R}^3} \psi^*_{k+}(\mathbf{r})\psi_{k-}(\mathbf{r})e^{-i\mathbf{p}\cdot\mathbf{r}}d\mathbf{r} \tag{13}$$

where $\psi_{k\pm}(\mathbf{r})$ are the coordinate space wavefunctions of the annihilating electron-positron pair.[2] The origin of Equation (13) is the two-step perturbation theory derivation of the annihilation probability, which in turn is based on the interaction of a Dirac electron with the quantized radiation field provided that, (a) the wavefunctions of the intermediate (virtual) states of the process are treated in the free particle approximation, and (b) the wavefunctions of the initial and final states are treated non-relativistically. Condition (b) implies that the initial and final states are represented by the Schrödinger wavefunctions multiplied by the

[2]The integrals in this section are over all electronic and positronic coordinates. For simplicity, the set notation on the integrals will be omitted going forward.

appropriate four-component column vector,

$$\psi_{k-}(\mathbf{r}) \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \end{vmatrix} \quad \text{and} \quad \psi_{k+}(\mathbf{r}) \begin{vmatrix} 0 \\ 0 \\ 1 \\ 0 \end{vmatrix} \tag{14}$$

which represent the electronic and positronic solutions to the Dirac four-component wave equation. Obviously, this formulation assumes the electron and positron are described by free particle wave functions. For our purposes, we need to be able to include electron-electron as well as electron-positron correlation in the calculation of our wavefunctions.

In 1956, Ferrell [17] presented the second-quantization formulation of the theory, which is also laid out by Mijnarends *et al.* [36] Also, Sakurai [48] presents an even more complete, covariant formulation; however, Ferrell's derivation is followed here. The decay of a low-energy electron-positron pair is a second-order process which proceeds via an intermediate state in which both particles and one photon are present. In view of the large momenta and energies of the annihilation quanta (and hence the large recoil of the scattered particle in the intermediate state) the transition matrix element is, to a good approximation, independent of the (small) initial momentum of the scattered particle; the same holds for the transition from the intermediate to the final state. Thus, as is customary in quantum electrodynamics, we employ completeness and sum over the intermediate states and the operator, producing a final state containing two photons with total momentum $\mathbf{p}$,

$$H'(\mathbf{p}) = \sum_{\mathbf{p}_1} \sum_{\mathbf{p}_2} a(\mathbf{p}_1) b(\mathbf{p}_2) \delta_{\mathbf{p}_1+\mathbf{p}_2,\mathbf{p}} \tag{15}$$

where $a(\mathbf{p}_1)$ and $b(\mathbf{p}_2)$ are the electron and positron annihilation operators, and $\delta_{\mathbf{p}_1+\mathbf{p}_2,\mathbf{p}}$ is a Dirac delta function. With the aid of the position-space field operators $\psi_-(\mathbf{r}_1)$ and $\psi_+(\mathbf{r}_2)$

defined by

$$a(\mathbf{p}_1) = V^{-1/2} \int e^{-i\mathbf{p}_1\cdot\mathbf{r}_1} \psi_-(\mathbf{r}_1) d\mathbf{r}_1 \tag{16}$$

$$b(\mathbf{p}_2) = V^{-1/2} \int e^{-i\mathbf{p}_2\cdot\mathbf{r}_2} \psi_+(\mathbf{r}_2) d\mathbf{r}_2 \tag{17}$$

where V is the quantization volume, the operator

$$H'(\mathbf{p}) = \int e^{-i\mathbf{p}\cdot\mathbf{r}} \psi_-(\mathbf{r}) \psi_+(\mathbf{r}) d\mathbf{r} \tag{18}$$

may be formed.

The application of $H'(\mathbf{p})$ to an initial electronic wave function that is a Slater determinant of orthonormal one-electron wave functions yields as many terms as there are electrons. Each term is a Slater determinant of next lower degree and is multiplied by a numerical factor identical to

$$\int e^{-i\mathbf{p}\cdot\mathbf{r}} \phi^e(\mathbf{r}) \phi^p(\mathbf{r}) d\mathbf{r} \tag{19}$$

where the annihilators in Equation (18) are replaced by the "wavefunction product," which signifies the electron and positron wavefunctions evaluated at the same point. This leads to the conclusion that the momentum distribution of the two-photon states, $\rho^{2\gamma}(\mathbf{p})$, arising from the annihilation of the electron-positron pair is obtained by squaring the absolute value of the Fourier transform of the "wavefunction product":

$$\rho^{2\gamma}(\mathbf{p}) = N \left| \int e^{-i\mathbf{p}\cdot\mathbf{r}} \phi^e(\mathbf{r}) \phi^p(\mathbf{r}) d\mathbf{r} \right|^2 \tag{20}$$

where N is a normalization constant to be determined later. [17] The $\rho^{2\gamma}(\mathbf{p})$ function is often referred to as the two-photon momentum density (TPMD) in the literature.

The total annihilation rate is obtained by taking the matrix element of $H'(\mathbf{p})$ between the initial and final states, squaring the absolute value, and summing over all final states of the matter system as well as over all values of $\mathbf{p}$. The result is proportional to the expectation

value, over the initial state, of the operator

$$\int (\psi_-^*(\mathbf{r})\psi_-(\mathbf{r})) \cdot (\psi_+^*(\mathbf{r})\psi_+^*(\mathbf{r}))d\mathbf{r}. \tag{21}$$

This is simply the electron density operator at the position of the positron, averaged over the positron position. Thus,

$$\lambda = \tau^{-1} = N \int n_-(\mathbf{r})n_+(\mathbf{r})d\mathbf{r} \tag{22}$$

where $n_{-(+)}(\mathbf{r})$ denotes the electron (positron) density. [17] In a homogeneous electron gas of density n_- the annihilation rate is

$$\lambda = \pi r_0^2 c n_- \tag{23}$$

where $r_0 = q^2/m_0c^2$ is the classical electron radius, and c the speed of light. This fixes the constant, N, to $\pi r_0^2 c$. [36]

2.2.2 Calculation of $\rho^{2\gamma}$ and λ in Condensed Matter. Several methods that were originally developed for electronic structure calculations have been adapted to calculate the positron distributions and related observables in periodic systems. This has been successful for simple metallic systems, where quantitative lifetime calculations yield results that agree with experiment to within a few picoseconds. [56] Chang Lee [34] investigated the nature of various approximations commonly made in the evaluation of $\rho^{2\gamma}$ and λ made in solid state calculations. He has shown that the Coulomb interaction between the electrons may be treated separately from the interaction leading to the annihilation process.[3] Furthermore, the reduction of the annihilation process in an n-electron system, described by a determinantal wavefunction, to a one-electron problem is shown to be equivalent to neglecting the possibility that the annihilation of an electron from a specific state is accompanied by the transition of another electron into this state, *i.e.*, the wavefunctions of the other electrons are

[3]For annihilation of Ps, the shift for the ground state energy is only 360 μeV, and the broadening due to the coupling with $|f\rangle$ is no more than $\sim \hbar/125$ ps $\approx 5\mu$eV. [52]

not allowed to change in the annihilation process. This amounts to reducing Equation (20) to a sum over all N_o occupied electron states:

$$\rho^{2\gamma}(\mathbf{p}) = N \sum_{j=1}^{N_o} \left| \int e^{-i\mathbf{p}\cdot\mathbf{r}} \phi_j^e(\mathbf{r}) \phi^+(\mathbf{r}) d\mathbf{r} \right|^2 . \tag{24}$$

In Equation (24), we have essentially ignored all possible correlation effects. This form of $\rho^{2\gamma}$, or the TPMD, is the foundation for most of the existing theoretical momentum density work in solids, and is usually referred to as the independent particle model (IPM). [36]

The zero positron density limit (ZPDL) is the most frequently used theoretical approach for treatment of positrons in materials. It relies on the fact that the electronic structure of the studied system is not influenced by the presence of positrons provided that the positron density is negligible everywhere in the system. This approximation is exactly valid for delocalized positrons in infinite defect-free systems. For a trapped (localized) positron, this approximation is not fulfilled, but still the ZPDL represents in many cases a reasonable approximation. In most applications of the ZPDL, the electronic structure of the studied system is first determined, then the positron potential is calculated as a sum of two parts

$$V_+(\mathbf{r}) = -V_{coul}(\mathbf{r}) + V_{corr}[n_-(\mathbf{r})]. \tag{25}$$

Several forms of $V_{corr}[n_-(\mathbf{r})]$ are used for practical calculations. Using this potential, the Schrödinger eigenvalue equation for positrons is solved in order to find the positron energy and wave function, and it is assumed that positrons occupy the lowest energy level due to fast thermalization. [56]

The form of the potential most commonly used in solid state electronic structure calculations, the linear-muffin-tin orbital (LMTO), is well suited for electrons, but it is less suitable for describing positron distributions in the interstitial region. Although it works well for bulk properties, such as positron affinity, it is especially problematic for open structures that are difficult to treat in the atomic sphere approximation (ASA). [4]

A group at Lawrence Livermore National Laboratory (LLNL) has modeled positron distributions in metals and alloys using a finite element-based approach, which combines the favorable properties of basis-set and real-space-grid approaches. Their method is capable of treating very large systems of thousands of atoms with arbitrary accuracy. Also, the full-potential approach makes no approximations to the shape of the positron potential, in contrast to the ASA-based approaches. The solutions to the Schrödinger equation are described in terms of strictly local, piecewise polynomial basis functions, and the unit cell is partitioned into subdomains called elements. The accuracy of this representation can be improved systematically by increasing the order of the polynomials or the number of elements–all the advantages and flexibility of a variational basis-set method. Also, finite element methods by nature are very scalable. [56]

The application of the finite element method requires a positron potential in order to calculate the positron wavefunction, and this potential is obtained by Sterne *et al.* [56] from overlapping atomic calculations. The electrostatic interactions with the electrons and the nuclei provide the dominant part of the potential, and this is obtained by overlapping atomic electrostatic potentials. The remaining electron-positron correlation potential is taken from a local density approximation (LDA) form where the potential depends only on the electron charge density, also obtained by overlapping atomic charge densities. So far, the method lacks electron-positron charge self-consistency (planned for future). The method compares well to LMTO-ASA calculations of metals, where they looked at positron lifetimes in metals and a few monovacancies (Cu, Al, Ag). They also applied the method to positrons in potassium-doped fullerenes. [56]

As mentioned previously, the calculation of the TPMD is the basis for most theoretical work in positron annihilation spectroscopy; however, when the electron-electron and electron-positron correlation effects are taken into account, the interaction of the particles with the lattice is extremely complicated. For this reason, the problem is usually discussed in terms of an LDA, as in the LLNL finite element approach. In this approach, the electron density is assumed to vary slowly over the unit cell and can be considered locally constant. In other words, the problem is reduced to a single positron in a locally homogeneous gas

of free electrons. This is a reasonable approach for metals, but breaks down in situations where the positron detaches itself from its screening electron cloud (a positron trapped in the image potential at a surface) or where the spatial extent of the screening cloud is comparable to that of the positron state (a positron trapped in a defect). For these situations, a two-component density-functional theory which includes the local response of the electron gas to the perturbing localized e^+ charge leads to two coupled Schrödinger-type equations which must be solved self-consistently. In this formalism, the pile-up of electron charge results in an *enhancement factor*, $\gamma[n_-(\mathbf{r})]$, which must be included in the calculation of λ and $\rho^{2\gamma}$ [36]

$$\lambda = \tau^{-1} = \pi r_0^2 c \int n_-(\mathbf{r}) n_+(\mathbf{r}) \gamma[n_-(\mathbf{r})] d\mathbf{r} \tag{26}$$

$$\rho^{2\gamma}(\mathbf{p}) = \pi r_0^2 c \sum_{j=1}^{N_o} \left| \int e^{-i\mathbf{p}\cdot\mathbf{r}} \phi_j^e(\mathbf{r}) \phi^+(\mathbf{r}) \gamma[n_j(\mathbf{r})] d\mathbf{r} \right|^2 . \tag{27}$$

2.2.3 Including Electron-Positron Correlation. Two main hurdles to overcome in calculating $\rho^{2\gamma}$ and λ for positrons in condensed matter are: (1) the inclusion of electron-electron and electron-positron correlation effects in the calculation of ϕ^e and ϕ^p, and (2) the inclusion of relaxation of the nuclear coordinates around the localized positron. In this section, methods for handling problem (1) are discussed.

D. M. Schrader published an excellent review of the knowledge of chemical compounds that contain positrons or Ps, in which he covers some elementary physics and establishes the kind of quantum mechanics (QM) that can be profitably applied to mixed electron-positron systems. [52] The methods that could be considered for solving the electron-positron correlation problem are:

- SVM
- QMC
- "traditional" quantum chemistry methods (*e.g.*, HF and CI)

Also, all of the binding energies and annihilation rates that are known for atomic and molecular systems are listed and discussed. It is noted that for atomic systems, most attention has been paid to the families on the left and right edges of the periodic table, leaving the interesting middle part less well understood.

The fact that the positron has the mass of an electron but the charge of a proton has serious consequences for quantum chemists trying to calculate mixed electron-positron wave functions: (a) the familiar Born-Oppenheimer approximation cannot be used for positrons, which must be treated as distinguishable electrons; (b) electron-positron correlation is more important, pair by pair, than electron-electron correlation; and (c) there are always core electrons (except for the simplest systems), but positrons, repelled by nuclei, congregate in the valence region and beyond. There are three interesting experimental observables that can be calculated for mixed electron-positron systems: binding energy, annihilation rate, and the TPMD. Both the annihilation rate and the TPMD have already been discussed. A rigorous definition of the binding energy is that a mixed electron-positron system is chemically stable in some state if the annihilation rate is greater than the sum of all other processes that depopulate that state. [52]

Since the positron has the same spin magnetic moment (except for direction) as the electron, familiar tools such as the Born-Oppenheimer approximation for nuclei and Russell-Saunders coupling for lepton spin in light atoms are appropriate in applications to electron-positron systems. In other words, after straightforward adjustment to accommodate two kinds of electrons, the full, formidable technology of quantum chemistry is available; however, for accurate results, electron-positron correlation must be accounted for, so single-particle methods such as SCF and many-body perturbation theory are not directly useable.

Of the three possible ways of including electron-positron correlation listed above, the "traditional" HF, CI, and multiconfiguration self-consistent field (MCSCF) quantum chemistry methods seem to hold the most promise for positrons in condensed matter. Due to the large number of basis functions needed for accurate calculations, only up to six

particles have been treated by the SVM. The QMC method is ideally suited for mixed systems because electron-positron correlation, which is difficult to treat with CI methods, is automatically treated correctly; however, systems of only about 10 leptons are routinely treated. Also, nonlocal operators, such as those for calculating the annihilation rate, are problematic.

The HF-based methods that have been adapted to mixed electron-positron systems are outlined in Section 2.2.5. The single-configuration HF method gives good binding energies for simple polar molecules, such as diatomic molecules, for which the bonding is predominately ionic, but a proper treatment of electron-positron correlation is required for accuracy in calculated annihilation rates. This correlation can be obtained with a CI approach, where the mixed electron-positron wavefunction is a linear combination of electron and positron determinants of spin orbitals.

2.2.4 Qualitative Description of the Positronic Wavefunction. If an atom's IP is below the ionization energy of Ps (6.803 eV or 0.25 a.u.), the system is described well by a positron bound to the neutral atom (e^+X). Conversely, if the IP is larger than 6.803 eV, the wavefunction resembles a positronium atom bound to a cation ($Ps[X^+]$). For atoms with IPs near 6.803 eV, there is expected to be an accidental degeneracy between the e^+X and $Ps[X^+]$. In this case, the atom will have a large positron affinity. This prediction is confirmed by the known data for neutral atoms. [52]

2.2.5 Nuclear-Electronic Orbital Approach. Hammes-Schiffer *et al.* have developed the NEO method [24, 44, 71] that is able to calculate mixed nuclear-electronic wave functions. In this method, both the electronic and nuclear molecular orbitals are expressed as linear combinations of Gaussian basis functions [see Equation (81) on page 75]. The advantages to this approach include:

- nuclear quantum effects are incorporated during electronic structure calculations
- the Born-Oppenheimer separation of electrons and nuclei is avoided
- excited vibrational-electronic states may be calculated

- its accuracy may be improved systematically
- incorporated into GAMESS quantum chemistry code, which is well-tested
- quantum tunneling is handled with multiple centers for a quantum nucleus
- positions of nuclear basis function centers may be fixed or optimized variationally
- symmetry-adapted linear combinations of nuclear molecular orbitals are available

In theory, within the NEO framework, the only difference between a quantum proton and a positron is the mass, and the method is able to model positronic systems after some modifications.

2.2.5.1 NEO-HF. In the following discussion, the NEO-HF method is described, comparing it to "conventional" HF theory which only treats the electronic wavefunction quantum mechanically (referred to simply as HF). In the equations, the terms which are unique to the NEO-HF method will be identified with a box around the expression (*i.e.*, $\boxed{x}$). Hopefully, this approach will help bring out the differences and similarities between NEO-HF and HF theory.

In the NEO approach, the system is divided into three parts: N_e electrons, N_p quantum nuclei, and N_c classical nuclei. The total Hamiltonian operator, H_{tot}, for the quantum subsystem includes the kinetic energy operators for the electrons and quantum nuclei, as well as potential energy operators for electron-electron, electron-classical nuclei, and electron-quantum nuclei Coulombic interactions [71]

$$\begin{aligned}\hat{H}_{tot} = &-\sum_{i}^{N_e}\frac{1}{2}\nabla_i^2\boxed{-\sum_{i'}^{N_p}\frac{1}{2M_{i'}}\nabla_{i'}^2} - \sum_{A}^{N_c}\sum_{i}^{N_e}\frac{Z_A}{r_{iA}} \\ &\boxed{+\sum_{A}^{N_c}\sum_{i'}^{N_p}\frac{Z_A Z_{i'}}{r_{i'A}} + \sum_{i'}^{N_p}\sum_{j'>i'}^{N_p}\frac{Z_{i'}Z_{j'}}{r_{i'j'}}} \\ &+\sum_{i}^{N_e}\sum_{j>i}^{N_e}\frac{1}{r_{ij}}\boxed{-\sum_{i}^{N_e}\sum_{i'}^{N_p}\frac{Z_{i'}}{r_{i'i}}}. \end{aligned} \tag{28}$$

Here, the unprimed indices, i and j, refer to electrons, the primed indices, i' and j', refer to quantum nuclei, and the indices A and B refer to classical nuclei. The masses, charges, and distances are denoted by M, Z, and r, respectively, with the appropriate subscripts. The classical nuclei are still treated by the Born-Oppenheimer approximation, and the Coulombic interaction between them is calculated at the end. The one-particle terms are defined as [71]

$$h^e(i) = \frac{1}{2}\nabla_i^2 - \sum_A^{N_c} \frac{Z_A}{r_{iA}} \tag{29}$$

$$\boxed{h^p(i') = \frac{1}{2M_{i'}}\nabla_{i'}^2 - \sum_A^{N_c} \frac{Z_A Z_{i'}}{r_{i'A}}} \tag{30}$$

and the total Hamiltonian can then be expressed as

$$\begin{aligned}\hat{H}_{tot} = &\sum_i^{N_e} h^e(i) \boxed{+ \sum_{i'}^{N_p} h^p(i') + \sum_{i'}^{N_p} \sum_{j'>i'}^{N_p} \frac{Z_{i'} Z_{j'}}{r_{i'j'}}} \\ &+ \sum_i^{N_e} \sum_{j>i}^{N_e} \frac{1}{r_{ij}} \boxed{- \sum_i^{N_e} \sum_{i'}^{N_p} \frac{Z_{i'}}{r_{i'i}}}.\end{aligned} \tag{31}$$

At the HF level, the mixed nuclear-electronic wavefunction, Ψ_{tot} is approximated as a single product of electronic and nuclear determinants of spin orbitals [71]

$$\Psi_{tot}(\mathbf{r}^e \boxed{, \mathbf{r}^p}) = \Phi_0^e(\mathbf{r}^e) \boxed{\Phi_0^p(\mathbf{r}^p)}. \tag{32}$$

In the above equations, $\Phi_0^e(\mathbf{r}^e)$ and $\Phi_0^p(\mathbf{r}^p)$ are antisymmetrized wave functions (determinants of spin orbitals) representing the electrons and fermionic nuclei, respectively. Here, $\mathbf{r}^e$ and $\mathbf{r}^p$ represent the spatial coordinates of the electrons and nuclei, respectively. According to the variational principle, the "best" HF spin orbitals are those which minimize the electronic energy. In NEO-HF, the energy is minimized with respect to both the electronic

and nuclear molecular orbitals, and the general total energy expression is [71]

$$
\begin{aligned}
E = & \sum_{i}^{N_e} \left[\chi_i^e \middle| h^e \middle| \chi_i^e\right] + \frac{1}{2}\sum_{i}^{N_e}\sum_{j}^{N_e}\Big(\left[\chi_i^e\chi_i^e\middle|\chi_j^e\chi_j^e\right] - \left[\chi_i^e\chi_i^e\middle|\chi_j^e\chi_j^e\right]\Big) \\
& \boxed{+\sum_{i'}^{N_p}\left[\chi_{i'}^p\middle|h^p\middle|\chi_{i'}^p\right] + \frac{1}{2}\sum_{i'}^{N_p}\sum_{j'}^{N_p}\Big(\left[\chi_{i'}^p\chi_{i'}^p\middle|\chi_{j'}^p\chi_{j'}^p\right] - \left[\chi_{i'}^p\chi_{i'}^p\middle|\chi_{j'}^p\chi_{j'}^p\right]\Big)} \\
& \boxed{-\sum_{i}^{N_e}\sum_{i'}^{N_p}\left[\chi_i^e\chi_i^e\middle|\chi_{i'}^p\chi_{i'}^p\right]}
\end{aligned}
\tag{33}
$$

where χ_i^e and $\chi_{i'}^p$ denote spin orbitals for the electrons and quantum nuclei, respectively. As usual, the two-electron integrals are defined as

$$
\left[\chi_i^e\chi_i^e\middle|\chi_j^e\chi_j^e\right] = \int d\mathbf{x}_1 \int d\mathbf{x}_2 \chi_i^{e*}(1)\chi_j^e(1)\frac{1}{r_{12}}\chi_k^{e*}(2)\chi_l^e(2) \tag{34}
$$

where $\mathbf{x}$ indicates both the spatial and spin coordinates. The other two-particle integrals are defined analogously. [For notational simplicity, the spatial and spin coordinates are not explicitly included in equations such as Equation (34).]

The total energy expression for a restricted Hartree-Fock (RHF) treatment of the electrons and a high-spin open-shell treatment of the quantum nuclei[4] is given by [71]

$$
\begin{aligned}
E = & 2\sum_{i}^{N_e/2} h_{ii}^e + \sum_{i}^{N_e/2}\sum_{j}^{N_e/2}\left[2\left(\psi_i^e\psi_i^e\middle|\psi_j^e\psi_j^e\right) - \left(\psi_i^e\psi_j^e\middle|\psi_i^e\psi_j^e\right)\right] \\
& \boxed{+\sum_{i'}^{N_p} h_{i'i'}^p + \frac{1}{2}\sum_{i'}^{N_p}\sum_{j'}^{N_p}\left[2\left(\psi_{i'}^p\psi_{i'}^p\middle|\psi_{j'}^p\psi_{j'}^p\right) - \left(\psi_{i'}^p\psi_{j'}^p\middle|\psi_{i'}^p\psi_{j'}^p\right)\right]} \\
& \boxed{-2\sum_{i}^{N_e/2}\sum_{i'}^{N_p}\left(\psi_i^e\psi_i^e\middle|\psi_{i'}^p\psi_{i'}^p\right)}
\end{aligned}
\tag{35}
$$

[4] Analogous equations may be derived in the same manner as presented for a restricted open-shell Hartree Fock (ROHF) or unrestricted Hartree-Fock (UHF) treatment of the electrons. A low-spin treatment of two quantum nuclei can be formulated in a straightforward manner. A general treatment of low-spin quantum nuclei requires a CI approach.

where the integrals are defined as

$$h_{ij}^{e} = \int d\mathbf{r}_1 \psi_i^{e*} h^e(1) \psi_j^e(1) \tag{36}$$

$$\boxed{h_{i'j'}^{p} = \int d\mathbf{r}_1 \psi_{i'}^{p*} h^p(1) \psi_{j'}^p(1)} \tag{37}$$

$$\left(\psi_i^e \psi_j^e \middle| \psi_k^e \psi_l^e\right) = \int d\mathbf{r}_1 \int d\mathbf{r}_2 \psi_i^{e*}(1) \psi_j^e(1) \frac{1}{r_{12}} \psi_k^{e*}(2) \psi_l^e(2) \tag{38}$$

and analogously for two-particle integrals involving the quantum nuclei. Here, the spatial molecular orbitals for the electrons and quantum nuclei are denoted ψ_i^e and $\psi_{i'}^p$, respectively.

The Hartree-Fock equations can then be derived using the standard variation method to minimize the energy in Equation (35) with respect to both the electronic and nuclear molecular orbitals

$$\psi_i^e(1) = \sum_{\mu}^{N_{bf}^e} c_{\mu i}^e \phi_\mu^e(1) \tag{39}$$

$$\boxed{\psi_{i'}^p(1) = \sum_{\mu'}^{N_{bf}^p} c_{\mu' i'}^p \phi_{\mu'}^p(1)}. \tag{40}$$

Here, the unprimed indices (μ, ν, σ, and λ) refer to electronic basis functions, and the primed indices refer to nuclear basis functions. This leads to the Hartree-Fock-Roothaan equations [71]

$$\sum_{\nu}^{N_{bf}^e} F_{\mu\nu}^e c_{\nu i}^e = \varepsilon_i^e \sum_{\nu}^{N_{bf}^e} S_{\mu\nu}^e c_{\nu i}^e \quad i = 1, \ldots, N_e/2 \tag{41}$$

$$\boxed{\sum_{\nu'}^{N_{bf}^p} F_{\mu'\nu'}^p c_{\nu' i'}^p = \varepsilon_{i'}^p \sum_{\nu'}^{N_{bf}^p} S_{\mu'\nu'}^p c_{\nu' i''}^p \quad i' = 1, \ldots, N_p}. \tag{42}$$

The Fock matrix elements may be expressed as [71]

$$F_{\mu\nu}^e = h_{\mu\nu}^e + G_{\mu\nu}^e \boxed{- \sum_{\lambda'\sigma'}^{N_{bf}^p} P_{\lambda'\sigma'}^p \left(\phi_\mu^e \phi_\nu^e \middle| \phi_{\sigma'}^p \phi_{\lambda'}^p\right)} \tag{43}$$

$$\boxed{F^p_{\mu'\nu'} = h^p_{\mu'\nu'} + G^p_{\mu'\nu'} - \sum_{\lambda\sigma}^{N^e_{bf}} P^e_{\lambda\sigma}\left(\phi^p_{\mu'}\phi^p_{\nu'}\middle|\phi^e_{\sigma}\phi^e_{\lambda}\right)} \tag{44}$$

where the density matrix elements are defines as

$$P^e_{\lambda\sigma} = 2\sum_{i}^{N_e/2} c^e_{\lambda i} c^{e*}_{\sigma i} \tag{45}$$

$$\boxed{P^p_{\lambda'\sigma'} = 2\sum_{i'}^{N_p} c^p_{\lambda' i'} c^{p*}_{\sigma' i'}} \tag{46}$$

and the standard one- and two-particle parts are defined as

$$h^e_{\mu\nu} = \int d\mathbf{r}_1 \phi^{e*}_{\mu} h^e(1) \phi^e_{\mu}(1) \tag{47}$$

$$\boxed{h^p_{\mu'\nu'} = \int d\mathbf{r}_1 \phi^{p*}_{\mu'} h^p(1) \phi^p_{\mu'}(1)} \tag{48}$$

$$G^e_{\mu\nu} = \sum_{\lambda\sigma}^{N^e_{bf}} P^e_{\lambda\sigma}\left[\left(\phi^e_{\mu}\phi^e_{\nu}\middle|\phi^e_{\sigma}\phi^e_{\lambda}\right) - \frac{1}{2}\left(\phi^e_{\mu}\phi^e_{\sigma}\middle|\phi^e_{\nu}\phi^e_{\lambda}\right)\right] \tag{49}$$

$$\boxed{G^p_{\mu'\nu'} = \sum_{\lambda'\sigma'}^{N^p_{bf}} P^p_{\lambda'\sigma'}\left[\left(\phi^p_{\mu'}\phi^p_{\nu'}\middle|\phi^p_{\sigma'}\phi^p_{\lambda'}\right) - \frac{1}{2}\left(\phi^p_{\mu'}\phi^p_{\sigma'}\middle|\phi^p_{\nu'}\phi^p_{\lambda'}\right)\right]}. \tag{50}$$

The overlap matrix elements are defined as [71]

$$S^e_{\mu\nu} = \int d\mathbf{r}_1 \phi^{e*}_{\mu} \phi^e_{\mu}(1) \tag{51}$$

$$\boxed{S^p_{\mu'\nu'} = \int d\mathbf{r}_1 \phi^{p*}_{\mu'} \phi^p_{\mu'}(1)}. \tag{52}$$

The form of the Fock matrix elements in Equations (43) and (44) clearly illustrates that the electronic and nuclear Fock equations are coupled to each other. Specifically, the electronic Fock matrix depends on the electronic density matrix through the second term and on the nuclear density matrix through the last term, while the nuclear Fock matrix

depends on the nuclear density matrix through the second term and on the electronic density matrix through the last term. The electronic and nuclear Hartree-Fock-Roothaan equations can be solved iteratively to self-consistency utilizing convergence accelerators previously developed for electronic structure theory. In the calculations of Webb *et al.* , [71] the nuclear Fock equations are fully converged after each step in the iterative procedure for the solution of the electronic Fock equations.

2.2.5.2 Nuclear-Electronic Correlation with NEO. In order to account for correlation effects, NEO-CI and a NEO-MCSCF methods are also implemented. In the NEO-CI method, the mixed nuclear-electronic wave function is a linear combination of products of electronic and nuclear determinants of spin orbitals:

$$\Psi_{tot}(\mathbf{r}^e, \mathbf{r}^p) = \sum_{I}^{N_{CI}^e} \sum_{I'}^{N_{CI}^p} C_{II'} \Phi_I^e(\mathbf{r}^e) \Phi_{I'}^p(\mathbf{r}^p). \tag{53}$$

The energy is minimized with respect to the CI coefficients, $C_{II'}$. In the NEO-MCSCF methodology, the mixed nuclear-electronic wave function is a linear combination of products of electronic and nuclear determinants of spin orbitals, and the energy is minimized with respect to the electronic and nuclear molecular orbitals as well as the CI coefficients. [71] Dynamical nuclear-electronic correlation is included in the NEO method with second-order perturbation theory (NEO-MP2). The details about how NEO was modified for positronic systems are given in Section 3.1 and Appendix A.

III. Modeling Positrons in Molecular Electronic Structure Calculations with the Nuclear-Electronic Orbital Method

In this chapter, the modifications that were necessary to treat positrons with the NEO method are discussed. The modified NEO code is applied to the PsH and e^+LiH systems, and positronic basis sets for PsH and e^+LiH are developed. The utility of the NEO method for modeling positronic systems is illustrated by computing the PEC for e^+LiH at the NEO-MP2 level.

3.1 Theory and Computational Method

3.1.1 Nuclear-Electronic Orbital Formulation. As discussed in Section 2.2.5, the original NEO method, which is incorporated into a recent version of the GAMESS code, [49] includes nuclear quantum effects directly within electronic structure calculations. It has been used mainly to treat protons quantum mechanically. For the special case of systems with N_e paired electrons, a single quantum proton, and N_c classical nuclei, previous studies [59, 60] illustrate that NEO-MP2 calculations with a NEO-HF reference Hamiltonian, defined as

$$\begin{aligned}\hat{H} = \; & -\sum_{i}^{N_e}\frac{1}{2}\nabla_i^2 - \frac{1}{2}\nabla_{1'}^2 - \sum_{A}^{N_c}\sum_{i}^{N_e}\frac{Z_A}{r_{iA}} \\ & + \sum_{A}^{N_c}\frac{Z_A}{r_{1'A}} + \sum_{i}^{N_e}\sum_{j>i}^{N_e}\frac{1}{r_{ij}} - \sum_{i}^{N_e}\frac{1}{r_{i1'}},\end{aligned} \tag{54}$$

capture more electron-proton correlation than analogous calculations with a reference Hamiltonian that includes the proton-proton interaction term. This trend is also expected to be true for electron-positron correlation. In equation (54), the unprimed indices, i and j, refer to electrons, the primed indices, i' and j', refer to quantum protons, and the index, A, refers to classical nuclei. The charges and distances are denoted by Z and r, respectively, with the appropriate subscripts. Takatsuka and Ten-no [65] used the Hamiltonian in Equation (54) in their MP2 calculations of PsH and PsF.

In principle, modifying the NEO method in GAMESS for the calculation of positronic systems is as simple as substituting the proton's mass with that of the positron. Since the initial implementation of NEO in GAMESS was developed for modeling quantum effects of nuclei, however, the original code did not allow nuclear basis functions to be centered on classical nuclei, where the electronic basis functions are located. Due to the diffuse nature of the positronic wavefunction, the optimal configuration for positronic basis functions is to be centered on classical nuclei with the electronic basis functions. The code was modified to allow for this situation in the extension of the NEO method for the calculation of mixed positronic-electronic wavefunctions.

In the resulting modified NEO-HF method, the energy corresponding to the single-configurational mixed positronic-electronic wavefunction is minimized with respect to the MOs. In the NEO-CASSCF method, the energy is minimized with respect to the MOs, as well as the configuration interaction (CI) coefficients, including all possible CI configurations that result from the chosen positronic and electronic active spaces in an analogous manner to the existing electronic structure method CASSCF. Within the NEO-CASSCF framework, if all of the quantum particles (*i.e.*, electrons and positrons) in the system are active and the active space includes all of the available MOs, then the calculation is termed NEO-FCI. In this case, the energy is minimized with respect to only the CI coefficients. Dynamical electron-electron and electron-positron correlation effects are included in the NEO framework using second-order perturbation theory (NEO-MP2). The NEO-HF and NEO-CASSCF formulations are presented in [71], and the NEO-MP2 method is presented in [60].

3.1.2 Positron Basis Set Development. In addition to modifying the existing NEO code in GAMESS, new positronic and modified electronic basis sets are needed before applying NEO to a molecular positronic-electronic system. For PsH, the basis function exponents in [6s], [6s1p], [6s2p], [6s3p], [6s2p1d], and [6s3p1d] basis sets for both electrons and the positron were optimized at the NEO-FCI level, using the same size basis sets for the electronic and positronic wavefunctions. These particular basis set sizes were chosen

for direct comparison to results in [64], in which basis function exponents for both electrons and the positron were optimized using a fully variational CI method. The optimized electronic and positronic basis sets for PsH are then used to compute NEO-MP2 energies. The effect of the basis set size on the amount of electron-electron and electron-positron correlation energy captured is discussed in Section 3.2.

In addition, even-tempered electronic and positronic basis sets for Li and PsH, respectively, were determined at the MP2 and NEO-MP2 levels. Note that the variational theorem does not apply to the second-order perturbation energy, which is not an expectation value of the Hamiltonian. Nevertheless, as noted by Krishnan *et al.* , [32] this method is useful for including correlation in basis set development. In the even-tempered scheme, the radial functions of the primitives are chosen such that the kth exponent, $\zeta_{k,l}$, of the set of Gaussian primitives of symmetry type l is specified with the even-tempered parameters α_l and β_l^k by the equation

$$\zeta_{k,l} = \alpha_l \beta_l^k. \tag{55}$$

Starting with an even-tempered basis on H^- and Li consisting of six s-type Gaussian primitives [6s], α_s and β_s are optimized as a function of the MP2 energy using the QDFIT2 program. [50] The optimization procedure is similar to that of Schmidt and Ruedenberg in that Gaussian primitives of each type are systematically added to the basis set until the improvement in the MP2 energy falls below a specific threshold. This initial basis set is then expanded by adding s-functions, one at a time, and the α_s and β_s are re-optimized using the optimum parameters from the previous iteration as the initial guess. The shell is considered full when the MP2 energy decreases by less than 0.1 millihartree. This procedure is then repeated for p- and d-type Gaussian primitives, beginning with three functions in each of the higher angular momentum shells. All even-tempered parameters are optimized at each step. Thus, the final step involves a six-parameter optimization (*i.e.*, α_l and β_l for l = s, p, d) for each electronic basis set. In some instances where four or more even-tempered parameters are optimized, the potential energy surface is found to be very flat, and the QDFIT2 program is not successful in finding the minimum of the MP2 energy. For these cases, the Hooke-

Jeeves generalized pattern search algorithm, implemented in the GenOpt program, [73] is used to locate the optimum values for each of the α_l and β_l.

The even-tempered electronic and positronic basis sets for PsH are then optimized by starting with the electronic basis sets developed for H^- and an initial even-tempered positronic basis set consisting of three s-type Gaussian primitives [3s]. The positronic basis set size is then increased using the same methodology as prescribed above while the electronic basis set size is held fixed at the size optimized for H^-. All even-tempered parameters for both electrons and positrons are optimized at each step, ultimately resulting in a 12-parameter optimization (*i.e.*, α_l and β_l for l = s, p, d for both the electrons and the positron) to obtain the final mixed electronic-positronic basis sets. After the optimum positronic basis set size was found for PsH, the electronic basis set size was increased in each shell and the parameters were re-optimized to confirm that the electronic basis set size was still optimum. The resulting even-tempered basis sets were used to compute the optimized geometry and vibrational frequency of e^+LiH.

3.1.3 Annihilation Rate. Using wavefunctions obtained with the NEO method, electron-positron annihilation rates can be computed at the NEO-HF and NEO-MP2 levels. Neglecting three-photon annihilation, the electron-positron annihilation rate for a bound state wavefunction, Ψ_0, consisting of N_e electrons and a single positron is given by [21]

$$\lambda = \pi r_0^2 c \int \sum_{i=1}^{N_e} \delta(\mathbf{r}_i^e - \mathbf{r}^p) \left|\Psi_0(\mathbf{r}^e, \mathbf{r}^p)\right|^2 d\mathbf{r}^e d\mathbf{r}^p \tag{56}$$

where the integration is over all electronic coordinates and the positronic coordinate.

At the NEO-HF level, for N doubly occupied electronic MOs, Equation (56) simplifies to

$$\lambda^{HF} = 2\pi r_0^2 c \sum_{i=1}^{N} S_{ii1'1'} \tag{57}$$

where $S_{ijj'j'}$ is the four-center integral,

$$S_{ijj'j'} = \int \phi_i(\mathbf{r})\phi_j(\mathbf{r})\phi_{i'}(\mathbf{r})\phi_{j'}(\mathbf{r}) d\mathbf{r} \tag{58}$$

r_0 is the classical electron radius, c is the speed of light, and $\phi_{i(i')}$ is the $i(i')$th electron (positron) MO; again, unprimed (primed) indices denote electron (positron) MOs. Note that the spatial coordinates in the four MOs in equation (58) are the same because of the Dirac delta function in Equation (56) and that $S_{iji'j'}$ has units of a.u.$^{-3}$.

Within the NEO-MP2 framework, the first-order correction to the annihilation rate is

$$\lambda^{(1)} = -4\pi r_0^2 c \sum_{arr'} \frac{\langle a1'|rr'\rangle}{\varepsilon_a + \varepsilon_{1'} - \varepsilon_r - \varepsilon_{r'}} S_{ar1'r'} \tag{59}$$

where the summation is over the RHF (restricted Hartree-Fock) occupied and virtual electron MOs (a and r, respectively) and the virtual positron MOs (r'). $\langle ii'|jj'\rangle$ is the electron-positron Coulomb integral, and $\varepsilon_{i(i')}$ is the eigenvalue for the ith (i'th) electron (positron) MO. The NEO-MP2 annihilation rate is then given by $\lambda^{\mathrm{MP2}} = \lambda^{\mathrm{HF}} + \lambda^{(1)}$. The module for computing annihilation rates based on Equations (57)-(59) has been incorporated into NEO in GAMESS.

3.2 Results and Discussion

In a positronic-electronic system, both electron-electron and electron-positron correlation corrections have substantial effects on calculated energies and wavefunctions. For the PsH system, the amount of correlation calculated with the NEO-MP2 and NEO-FCI methods and with different basis set sizes was investigated. In Table 2, NEO-HF and NEO-MP2 energies and annihilation rates are given for PsH with [6s], [6s1p], [6s2p], [6s3p], [6s2p1d], and [6s3p1d] basis sets with exponents optimized at the NEO-FCI level. Also provided are the electron-electron and electron-positron correlation energies recovered with NEO-MP2. The total correlation energy recovered with NEO-MP2 is then compared to the NEO-FCI correlation energy.

Two significant trends are evident from the data in Table 2. As the basis set size is increased, (1) the magnitudes of the electron-electron and electron-positron correlation energies increase, providing virtually all of the improvement in the NEO-MP2 and NEO-FCI energies (*i.e.*, the change in the NEO-HF energy is negligible); and (2) the fraction of

Table 2: Comparison of NEO-MP2 and NEO-FCI results for PsH. NEO-HF, NEO-MP2, and NEO-FCI energies [E_{HF}, E_{MP2}, and E_{FCI}, respectively] and NEO-HF and NEO-MP2 annihilation rates [λ^{HF} and λ^{MP2}, respectively] were obtained for PsH with basis sets optimized at the NEO-FCI level. The correlation energy recovered with NEO-MP2 [$E^{\mathrm{corr}}_{\mathrm{MP2}} = E_{\mathrm{MP2}} - E_{\mathrm{HF}} = E^{\mathrm{ee}}_{\mathrm{MP2}} + E^{\mathrm{ep}}_{\mathrm{MP2}}$] and NEO-FCI [$E^{\mathrm{corr}}_{\mathrm{FCI}} = E_{\mathrm{FCI}} - E_{\mathrm{HF}}$], the second-order electron-electron [$E^{\mathrm{ee}}_{\mathrm{MP2}}$] and electron-positron [$E^{\mathrm{ep}}_{\mathrm{MP2}}$] corrections that comprise the NEO-MP2 correction, and the fraction of the correlation energy recovered with NEO-MP2 [$E^{\mathrm{corr}}_{\mathrm{MP2}}/E^{\mathrm{corr}}_{\mathrm{FCI}}$] are also provided.

	[6s]	[6s1p]	[6s2p]	[6s3p]	[6s2p1d]	[6s3p1d]
E_{HF}	−0.666766	−0.666791	−0.666783	−0.666784	−0.666865	−0.666872
E_{MP2}	−0.683835	−0.715107	−0.721801	−0.723411	−0.726725	−0.728306
E_{FCI}	−0.691010	−0.732176	−0.743336	−0.745807	−0.756408	−0.758965
$E^{\mathrm{ee}}_{\mathrm{MP2}}$	−0.010649	−0.020459	−0.024004	−0.025103	−0.025226	−0.026309
$E^{\mathrm{ep}}_{\mathrm{MP2}}$	−0.006420	−0.027858	−0.031014	−0.031525	−0.034634	−0.035125
$E^{\mathrm{corr}}_{\mathrm{MP2}}$	−0.017070	−0.048316	−0.055018	−0.056628	−0.059860	−0.061435
$E^{\mathrm{corr}}_{\mathrm{FCI}}$	−0.024244	−0.065385	−0.076553	−0.079023	−0.089544	−0.092094
$E^{\mathrm{corr}}_{\mathrm{MP2}}/E^{\mathrm{corr}}_{\mathrm{FCI}}$	0.704079	0.738948	0.718686	0.716599	0.668503	0.667083
λ^{HF} (ns^{-1})	0.297218	0.297585	0.297870	0.297660	0.297895	0.297736
λ^{MP2} (ns^{-1})	0.323138	0.380956	0.395822	0.400379	0.413305	0.417694

the correlation energy recovered with NEO-MP2 versus NEO-FCI decreases slightly. The increasing values for the electron-electron and electron-positron correlation energy seen in the first trend indicate that the larger basis set yields a better set of orbitals for recovery of dynamical correlation through perturbation methods. The increase in the electron-positron correlation energy corresponds to an increase in the electron-positron annihilation rate. The second trend illustrates that the increase in the NEO-MP2 correlation energy with basis set size is slower than the increase in the NEO-FCI correlation energy. The multiconfigurational character of the NEO-FCI wavefunction can be determined by examination of the natural orbital occupation numbers (NOON). The highest NOON value for the positron changes from 0.9872 for the [6s] basis set to 0.8837 for the [6s3p1d] basis, a decrease of 10.4 %. This decrease indicates that the NEO-FCI wavefunction becomes more multiconfigurational for this choice of orbitals as the basis set increases.

Even-tempered basis sets for H^-, Li, and PsH were optimized using the procedure outlined in section 3.1.2. The even-tempered parameters are given in Table 3. The opti-

Table 3: H^-, Li, PsH, and e^+LiH even-tempered basis set parameters and associated energies optimized with MP2 and NEO-MP2 methods. NEO-MP2 positron parameter optimizations for e^+LiH were carried out with and without an electronic frozen core (FC).

Basis	α_s	β_s	α_p	β_p	α_d	β_d	E_{MP2}
H^- e^- 9s5p4d	0.004082	3.12031	0.018186	2.74250	0.033809	2.77814	−0.516892
Li e^- 13s6p4d	0.010651	2.65645	0.066641	2.77175	0.329441	2.81975	−7.472392
PsH e^- 9s5p4d	0.018298	2.66531	0.029626	2.46850	0.041616	2.66534	−0.731866
e^+ 6s4p4d	0.012420	3.91161	0.029026	2.36631	0.022201	3.55911	
e^+LiH/FC e^+ 6s4p4d	0.009969	1.58471	0.004681	3.30361	0.013346	3.25261	−8.028839
e^+LiH e^+ 6s4p4d	0.007373	1.76822	0.004573	3.32930	0.016052	3.31927	−8.068859

mum electronic basis set sizes for H^- and Li were found to be [9s5p4d] and [13s6p4d], respectively, and the optimum positronic basis set size for PsH is [6s4p4d]. In general, the optimal electronic and positronic basis sets have approximately the same number of p- and d-type primitives, while the positronic basis set has fewer s-type primitives than the electronic basis. The NEO-MP2 energy with the resulting [9s5p4d-6s4p4d] basis set is −0.731866 Hartree, which is 0.00356 Hartree lower than the best NEO-MP2 result achieved with the FCI-optimized basis sets of Table 2; however, it is still 0.027099 Hartree higher than the best NEO-FCI result given.

Subsequent to this even-tempered basis set optimization, the PEC for e^+LiH was computed at the NEO-MP2 level with the Li and PsH even-tempered basis sets. The energy of e^+LiH decreased significantly when the even-tempered parameters for the positronic basis set on the hydrogen atom and the internuclear distance were simultaneously optimized to minimize the NEO-MP2 energy. The effect of using a frozen core for the Li atom was also investigated. Typically, a frozen core is not used for Li in electronic structure calculations, but previous studies [8] suggest that the balance in correlation energy for LiH and e^+LiH can be improved by restricting the active space of electrons to exclude the 1s shell for Li. With the 1s frozen core on Li, the dissociation energy is 0.537992 eV, which is much closer to the SVM result of 0.555405 eV [39] than the all-electron calculation. The resulting e^+LiH positronic basis sets with and without the frozen core are provided in Table 3.

Table 4: LiH and e^+LiH MP2 and NEO-MP2 results. The Hartree-Fock energies [$E_{\rm HF}$], MP2 energies [$E_{\rm MP2}$], and electron-electron and electron-positron correlation contributions to the MP2 energies ($E^{\rm ee}_{\rm MP2}$ and $E^{\rm ep}_{\rm MP2}$, respectively) are provided. Also given are the bond lengths (R_e and R_0), dissociation energies (Δ), and vibrational energies (ν). Calculations with a frozen 1s electron orbital on Li are denoted with FC. Energies are in Hartrees unless otherwise specified.

	$E_{\rm HF}$	$E^{\rm ee}_{\rm MP2}$	$E^{\rm ep}_{\rm MP2}$	$E_{\rm MP2}$	R_e(Å)	R_0(Å)[a]	Δ (eV)[b]	ν (cm^{-1})[a]
LiH/FC	−7.987240	−0.028657	-	−8.015897	1.6006	1.6088	-	1400
e^+LiH/FC	−7.991726	−0.028425	−0.008688	−8.028839	1.6799	1.6835	0.537992	1146
LiH	−7.987210	−0.069001	-	−8.056211	1.5892	1.5979	-	1410
e^+LiH	−7.991805	−0.068397	−0.008657	−8.068859	1.6652	1.6691	1.625510	1164

[a]From [9], the experimental LiH bond length and vibrational frequency are 1.5957 Å and 1406 cm^{-1}, respectively.

[b]The e^+LiH dissociation energy is computed for the $e^+LiH \rightarrow Li^+ + PsH$ dissociation channel, using an Li^+ MP2 energy of −7.274638 Hartree computed with the Li MP2-optimized [13s6p4d] even-tempered basis set and the PsH NEO-MP2 energy given in Table 3.

A summary of the results obtained with the even-tempered basis sets optimized for the molecular system is provided in Table 4. This table includes the NEO-HF and NEO-MP2 energies, Li-H bond lengths R_e and R_0, and vibrational energies (ν). Also given in the table are the dissociation energies (Δ) for the lowest-energy dissociation channel, $e^+LiH \rightarrow Li^+ + PsH$. These calculations were performed both with and without a frozen core (FC) on the Li atom. The values of R_e, R_0, and ν were determined by computing PECs along the Li-H distance, R_{LiH}, and fitting the data to a Morse potential, as depicted in Figure 2. The Morse potential was determined with a least squares method using the data points for 1.25 Å $< R_{\rm LiH} <$ 2.5 Å, and the eigenvalues and eigenfunctions of this Morse potential were computed analytically. [5] The values of R_0 were obtained from the average of R_{LiH} over the ground state vibrational wavefunction, and the vibrational energies were determined from the splitting between the lowest two vibrational states for this Morse potential. For all basis sets and levels of theory, the computed values for R_0 and ν of the non-positronic LiH molecule agree well with the experimental values of 1.5957 Å and 1406 cm^{-1}, respectively. [9] The values of R_0 and ν computed for e^+LiH with a frozen 1s core on the Li atom are 1.6835 Å and 1146 cm^{-1}, respectively. Thus, the vibrational energy of e^+LiH is computed to be approximately 30 meV less than that of LiH.

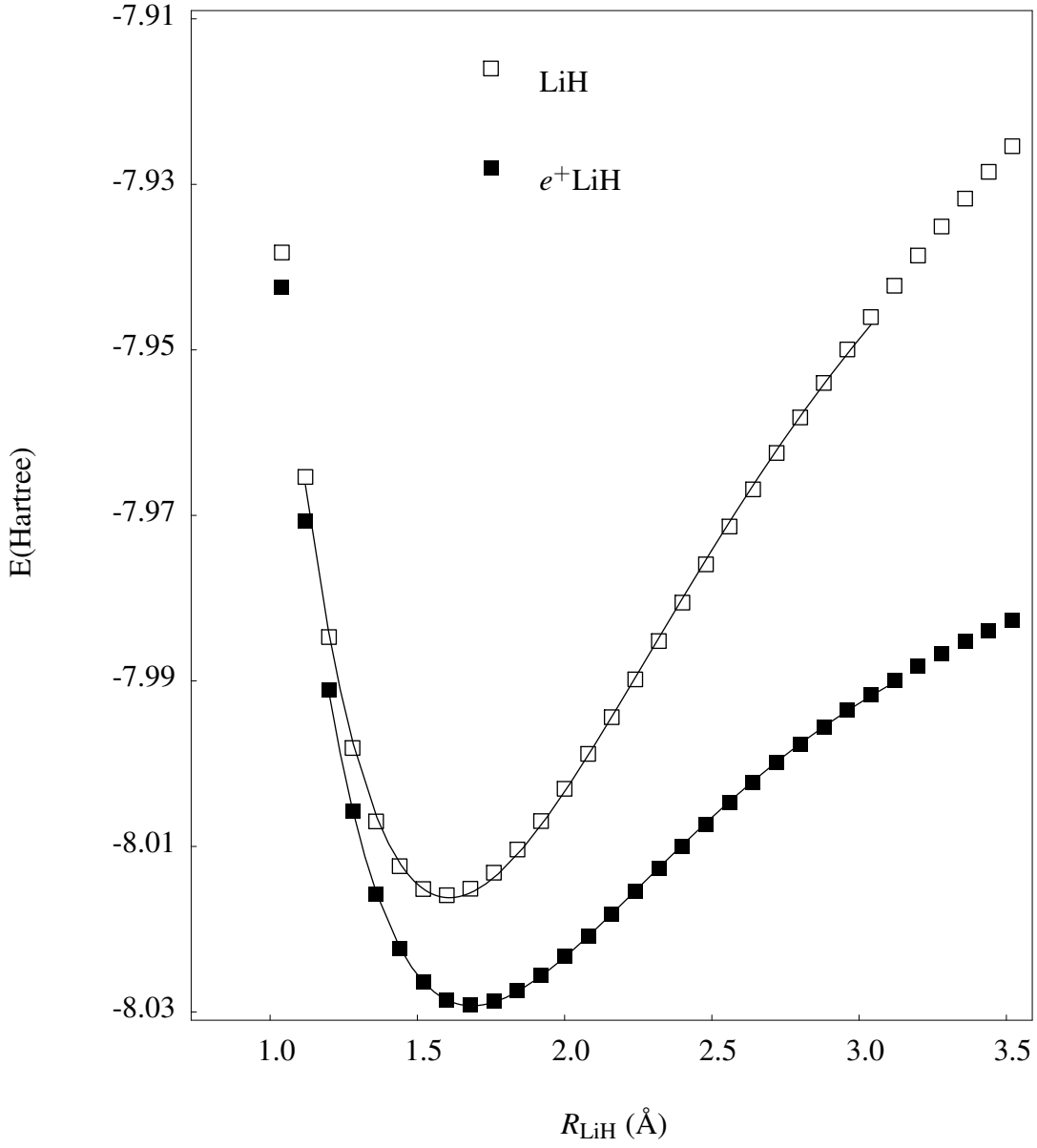

Figure 2: Comparison of LiH and e^+LiH MP2 and NEO-MP2 PECs, with a 1s frozen core on the Li atom and associated electronic and positronic basis sets from Table 3. The Morse potential fits used to compute the vibrational energies are also shown.

It is also important to consider whether positron basis functions should be centered on only the more electronegative H atom or on both the H and Li atoms in e^+LiH. To explore this issue, optimum electronic and positronic basis sets were found for LiPs using the same method that was applied to PsH above. Starting with the LiPs and PsH electronic and positronic basis sets, the positronic molecular basis set parameters and the internuclear distance for e^+LiH was determined at the NEO-MP2 level. The resulting minimum energy without a frozen [1s] core on the Li atom was -8.069102 Hartree, a decrease of only 0.000243 Hartree compared to the calculation with positron basis functions on only the H atom. Also, the PEC for the system was recalculated, and the equilibrium bond length and vibrational energy were 1.6673 Å and 1159 cm^{-1}, respectively, which are very close to the numbers previously obtained. From these results, one can conclude that for e^+LiH, placement of the positronic basis functions on only the more electronegative hydrogen atom is sufficient.

3.3 Conclusion

The NEO method is not yet a quantitative tool for computing energies and annihilation rates for positronic systems. With NEO, the best energy and annihilation rate for PsH are -0.758965 Hartree and 0.41769 ns^{-1}, respectively. Recently, Mitroy reported a PsH energy of -0.789196470 Hartree and an annihilation rate of 2.47178 ns^{-1} from nearly converged SVM calculations. [37] The methods presented here, however, provide a foundation upon which future general methods for studying positronic systems and modeling large positronic-electronic molecular systems can be built. Improvements to the quantitative nature of the method can be made by developing larger positron basis sets and including more electron-electron and electron-positron correlation energy.

The modified NEO approach is a potentially powerful method for computing mixed positronic-electronic wavefunctions. Its utility is demonstrated by computing the PEC of e^+LiH, allowing the prediction of its equilibrium geometry, vibrational energy, and dissociation energy. Also, the energy and the annihilation rate for PsH is calculated at the NEO-MP2 level. In addition, the amount of electron-electron and electron-positron corre-

lation energy captured with the NEO-MP2 and NEO-FCI methods for PsH are compared. Finally, a systematic method for developing even-tempered electronic and positronic basis sets for NEO calculations has been demonstrated. Basis sets developed using atomic calculations were found to provide a reasonable starting point for molecular calculations; however, further optimization of the positronic basis set parameters was necessary to accurately describe the PEC of the molecule.

3.4 Supporting Information

In addition to the Hamiltonian given in Equation (54), referred to in this section as the DIAGZN Hamiltonian, there is a second Hamiltonian available in the NEO method. This second Hamiltonian, referred to here as the RONHF Hamiltonian, is valid for N_e paired electrons, N_p unpaired high-spin quantum protons, and N_c classical nuclei. It includes the kinetic energy operators for the electrons and quantum protons and the various potential energy operators, including the proton-proton repulsion term

$$\begin{aligned}\hat{H} = & -\sum_{i}^{N_e}\frac{1}{2}\nabla_i^2-\sum_{i'}^{N_p}\frac{1}{2}\nabla_{i'}^2 \\ & -\sum_{A}^{N_c}\sum_{i}^{N_e}\frac{Z_A}{r_{iA}}+\sum_{A}^{N_c}\sum_{i'}^{N_p}\frac{Z_A}{r_{i'A}} \\ & +\sum_{i'}^{N_p}\sum_{j'>i'}^{N_p}\frac{1}{r_{i'j'}}+\sum_{i}^{N_e}\sum_{j>i}^{N_e}\frac{1}{r_{ij}}-\sum_{i}^{N_e}\sum_{i'}^{N_p}\frac{1}{r_{ii'}}. \end{aligned} \tag{60}$$

Here, the unprimed indices, i and j, refer to electrons, the primed indices, i' and j', refer to quantum protons, and the index, A, refers to classical nuclei. The charges and distances are denoted by Z and r, respectively, with the appropriate subscripts.

The NEO-MP2 (with the RONHF Hamiltonian) and NEO-FCI results for PsH are compared in Table 5. As in Table 2, the NEO-MP2 calculations were performed with the basis sets optimized at the NEO-FCI level. The ratios of the correlation energy captured with the NEO-MP2 method to the amount captured with NEO-FCI were computed for comparison with values obtained with the DIAGZN Hamiltonian. It is clear that for a

Table 5: Comparison of NEO-MP2 (with RONHF Hamiltonian) and NEO-FCI results for PsH. NEO-MP2 energies ($\tilde{E}_{\mathrm{MP2}}$) and annihilation rates ($\tilde{\lambda}^{\mathrm{MP2}}$) are obtained for PsH with basis sets optimized at the NEO-FCI level. The correlation energy recovered with NEO-MP2 [$\tilde{E}^{\mathrm{corr}}_{\mathrm{MP2}} = \tilde{E}_{\mathrm{MP2}} - E_{\mathrm{HF}}$], the second-order electron-positron correction [$\tilde{E}^{\mathrm{ep}}_{\mathrm{MP2}}$], and the fraction of the correlation energy recovered with NEO-MP2 [$\tilde{E}^{\mathrm{corr}}_{\mathrm{MP2}}/E^{\mathrm{corr}}_{\mathrm{FCI}}$] are also listed. The NEO-MP2 energies and annihilation rates listed here correspond to the RONHF Hamiltonian that includes the positron-positron Coulomb-exchange operator.

	[6s]	[6s1p]	[6s2p]	[6s3p]	[6s2p1d]	[6s3p1d]
$\tilde{E}^{\mathrm{ep}}_{\mathrm{MP2}}$	−0.0051922	−0.0226576	−0.0250056	−0.0254306	−0.0281093	−0.0285159
$\tilde{E}_{\mathrm{MP2}}$	−0.6826072	−0.7099069	−0.7157925	−0.7173172	−0.7202002	−0.7216970
$\tilde{E}^{\mathrm{corr}}_{\mathrm{MP2}}$	−0.0158413	−0.0431163	−0.0490094	−0.0505336	−0.0533355	−0.0548252
$\tilde{E}^{\mathrm{corr}}_{\mathrm{MP2}}/E^{\mathrm{corr}}_{\mathrm{FCI}}$	0.6534186	0.6594205	0.6402030	0.6394801	0.5956376	0.5953152
$\tilde{\lambda}^{\mathrm{MP2}}$ (ns^{-1})	0.3190397	0.3662093	0.3794839	0.3838186	0.3944266	0.3986460

Table 6: PsH and e^+LiH even-tempered basis set parameters and associated energies optimized at the NEO-MP2 level (with RONHF Hamiltonian). NEO-MP2 positron parameter optimizations for e^+LiH were carried out with and without an electronic frozen core (FC).

System	Basis	α_s	β_s	α_p	β_p	α_d	β_d	$\tilde{E}_{\mathrm{MP2}}$
PsH	e^- 9s5p4d	0.017642	2.64131	0.031256	2.45050	0.048099	2.48114	−0.724795
	e^+ 6s4p4d	0.010055	3.37831	0.025549	2.48931	0.048614	2.36731	
e^+LiH/FC	e^+ 6s4p4d	0.028535	2.79625	0.003695	3.30505	0.004343	3.10865	−8.027710
e^+LiH	e^+ 6s4p4d	0.008395	3.42195	0.003593	3.32195	0.004126	3.14285	−8.067716

given FCI-optimized basis set, NEO-MP2 with the DIAGZN Hamiltonian captures more electron-positron correlation energy than NEO-MP2 with the RONHF Hamiltonian. This increase in the electron-positron correlation energy corresponds to an increase in the computed electron-positron annihilation rate.

The PsH and e^+LiH even-tempered basis set parameters optimized at the NEO-MP2 level with the RONHF Hamiltonian are given in Table 6. The optimum basis set sizes are the same as those found with the DIAGZN Hamiltonian in Table 3, and the even-tempered basis set parameters are similar as well. The most significant difference is in the values of α_s and β_s for e^+LiH (both with and without the 1s frozen electronic core), in which the DIAGZN values are lower.

Table 7: LiH and e^+LiH MP2 and NEO-MP2 results (with RONHF Hamiltonian). The NEO-HF energies (E_{HF}), NEO-MP2 energies ($\tilde{E}_{MP2}$), and the electron-electron and electron-positron correlation contributions to the NEO-MP2 energies (E^{ee}_{MP2} and $\tilde{E}^{ep}_{MP2}$, respectively) are listed. Also listed are the equilibrium bond lengths (R_e), dissociation energies (Δ), and vibrational energies (ν). Calculations were accomplished both with and without a frozen 1s electronic core on the Li atom. Energies are in Hartrees unless otherwise specified.

	E_{HF}	E^{ee}_{MP2}	$\tilde{E}^{ep}_{MP2}$	$\tilde{E}_{MP2}$	R_e (Å)	Δ (eV)	ν (cm^{-1})
LiH/FC	−7.987229	−0.028659	-	−8.015888	1.6002	-	1401
e^+LiH/FC	−7.992352	−0.028438	−0.006920	−8.027710	1.6700	0.698130	1175
LiH	−7.987200	−0.069004	-	−8.056203	1.5891	-	1411
e^+LiH	−7.992461	−0.068441	−0.006814	−8.067716	1.6555	1.785350	1192

The MP2 and NEO-MP2 results for LiH and e^+LiH with the RONHF Hamiltonian are given in Table 7. Comparing to the results given in Table 4 for the DIAGZN Hamiltonian, we see that NEO-MP2 with the RONHF Hamiltonian gives less bond length relaxation and a smaller decrease in the vibrational energy with addition of the positron to LiH. Also, the calculated value for the dissociation energy with the frozen 1s electronic core on the Li atom does not agree with the SVM results as well.

In Figure 3, the variation of E^{ee}_{MP2} and E^{ep}_{MP2} as a function of the Li-H bond distance is plotted along with the E_{MP2} and E_{HF} energies. While E^{ee}_{MP2} is fairly constant with R_{LiH}, as the system dissociates into Li^+ and PsH, the magnitude of E^{ep}_{MP2} increases significantly and becomes equal to E^{ee}_{MP2}. This illustrates the importance of accuracy in the calculation of electron-positron correlation, especially as the system dissociates into Li^+ and PsH.

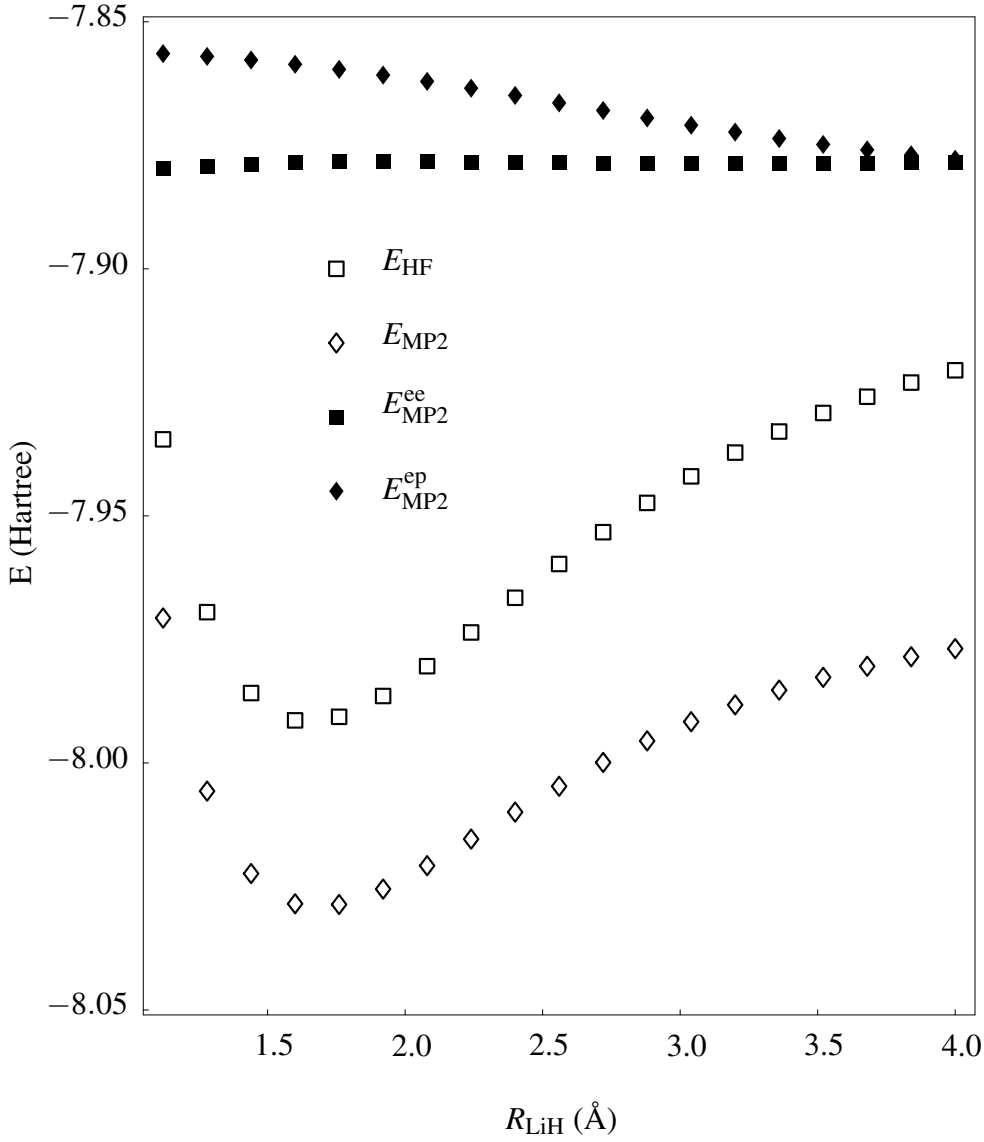

Figure 3: e^+LiH HF and MP2 PECs with the electron-electron and electron-positron correlation energies ($E^{\mathrm{ee}}_{\mathrm{MP2}}$ and $E^{\mathrm{ep}}_{\mathrm{MP2}}$, respectively). The DIAGZN Hamiltonian with a 1s frozen electronic core on the Li atom and basis sets from Table 3 were used. Electronic and positronic basis functions were centered on both the Li and H atoms. $E^{\mathrm{ee}}_{\mathrm{MP2}}$ and $E^{\mathrm{ep}}_{\mathrm{MP2}}$ are offset vertically for convenience.

IV. Vibrational Energy Levels of LiX and e^+LiX (X = H, F, Cl)

4.1 Introduction

In Equation (1), it is assumed that the shape of the PES of the positronic molecule is unchanged and simply shifted down by ε_b. Also, VFR which occur between vibrational energy levels other than $\upsilon = 0$ and $\upsilon = 1$ are neglected. In a more complete treatment of VFR theory, changes in the shape of the PES upon addition of the positron are allowed, and resonances between vibrational energy levels other than $\upsilon = 0$ and $\upsilon = 1$ are not excluded. For such a description, a resonance will occur when the incident positron energy plus the vibrational energy of the molecule is equal to the energy of a vibrationally excited molecule-positron complex

$$E_{\upsilon=n} + \varepsilon_k = E_{\upsilon=n'} \tag{61}$$

where $E_{\upsilon=n}$ is the energy of the non-positronic molecule in the $\upsilon = n$ energy level, and $E_{\upsilon=n'}$ is the energy of the positron-molecule complex in the $\upsilon = n'$ energy level. Note that in Equation (61), we do not require the non-positronic molecule to be in the ground vibrational state. Also, the resonance can occur for any cases where $E_{\upsilon=n'} > E_{\upsilon=n}$.

Thus, in order to calculate the possible VFR for a given molecule, one must compute the PES of both the non-positronic and the positronic molecule to account for the effect of the positronic-electronic potential on the vibrational structure. Likewise, if one can measure multiple VFR for a given molecule at different incident positron kinetic energies, it might be possible to determine the vibrational structure (and hence the PES) of the positronic molecule. Such fidelity would allow for a direct comparison of theory to experiment, not just an experimental determination of ε_b.

4.2 Theory

The energy of the $\upsilon = n$ energy level of a non-rotating diatomic molecule is given approximately by

$$E_{\upsilon=n} = \omega_e \left(n + \frac{1}{2}\right) - \omega_e \chi_e \left(n + \frac{1}{2}\right)^2 \tag{62}$$

in which the harmonic and anharmonic parameters are ω_e and $\omega_e\chi_e$, respectively. If one measures a VFR at a positron kinetic energy of ε_1, involving the $\upsilon = 0$ state of the non-positronic molecule and the $\upsilon = p'$ state of the positronic system, Equation (61) becomes

$$E_{\upsilon=0} + \varepsilon_1 = E_{\upsilon=p'}. \tag{63}$$

If one measures a second VFR at a positron kinetic energy of ε_2 that is identified as involving the $\upsilon = 0$ state of the non-positronic molecule and the $\upsilon = q'$ state of the positronic system, we get a second equation

$$E_{\upsilon=0} + \varepsilon_2 = E_{\upsilon=q'}. \tag{64}$$

Now, substituting Equation (62) for the vibrational energies, we get

$$\frac{1}{2}\omega_e - \frac{1}{4}\omega_e\chi_e + \varepsilon_1 = \omega'_e\left(p' + \frac{1}{2}\right) - \omega'_e\chi'_e\left(p' + \frac{1}{2}\right)^2 \tag{65}$$

and

$$\frac{1}{2}\omega_e - \frac{1}{4}\omega_e\chi_e + \varepsilon_2 = \omega'_e\left(q' + \frac{1}{2}\right) - \omega'_e\chi'_e\left(q' + \frac{1}{2}\right)^2 \tag{66}$$

Thus, if ω_e and $\omega_e\chi_e$ are known, and one can identify at least two transitions with a common vibrational energy level [on either side of Equation (61)], then ω'_e and $\omega'_e\chi'_e$ can be determined and the PEC for the positronic diatomic is known. Using these parameters, we are now able to compute the binding energy of the positron to the ground state of the positronic molecule

$$\begin{aligned} \varepsilon_b &= E_{\upsilon=0} - E_{\upsilon=0'} && (67)\\ &= \frac{1}{2}\omega_e - \frac{1}{4}\omega_e\chi_e - \left(\frac{1}{2}\omega'_e - \frac{1}{4}\omega'_e\chi'_e\right) && (68) \end{aligned}$$

4.3 Computational Method

In this chapter, PECs for LiX and e^+LiX (X = H, F, Cl) are computed for their ground electronic states, and vibrational energies (including zero point energy correction) are determined by fitting them to a Morse potential. The results are then used to predict the possible VFR for these molecules using Equation (61). All calculations are done at the MP2 level using the GAMESS electronic structure code. [49] The e^+LiX systems are modeled with the NEO method [60, 71], modified for the treatment of positrons as discussed in Chapter III.

Even-tempered electronic and positronic basis sets were developed for H^-, PsH, Li^-, and LiPs in Chapter III, and using the same methodology, basis sets for F^-, PsF, Cl^- and PsCl were developed. In the even-tempered scheme, the radial functions of the primitives are chosen such that the kth exponent, $\zeta_{k,l}$, of the set of Gaussian primitives of symmetry type l is specified with the even-tempered parameters, α_l and β_l^k, by the equation

$$\zeta_{k,l} = \alpha_l \beta_l^k \tag{69}$$

which are optimized versus the MP2 and NEO-MP2 energies.

It was shown in Section 3.2 that the positronic basis sets developed for atoms require further optimization when used in molecular systems. For the e^+LiX systems, this optimization was accomplished by simultaneously varying the α and β values for the positronic basis sets previously developed for the Li and X atoms along with the internuclear distance, R_{LiX}. The α and β values for the electronic basis sets are held fixed during the optimization. In the end, we are left with a positronic "molecular basis set" which consists of the same number of s-, p-, and d-functions as the LiPs and PsX basis sets, and we find the equilibrium geometry, R_e, for the e^+LiX system.

4.4 Results and Discussion

The even-tempered basis set parameters for X^-, PsX, Li^-, LiPs, LiX, and e^+LiX and their associated MP2 and NEO-MP2 energies are listed in Table 8. Using these basis

sets, PECs for the LiX and e^+LiX systems were computed at the MP2 and NEO-MP2 levels and are shown in Figure 4. The curves were fit to a Morse potential to determine the bond lengths, R_e and R_0, the harmonic and anharmonic parameters, and the vibrational energies. The first three vibrational energy levels ($\upsilon = 0$ through 2) for the non-positronic molecules, the first eight energy levels ($\upsilon = 0$ through 7) of e^+LiH and e^+LiF, and the first nine energy levels ($\upsilon = 0$ through 8) of e^+LiCl are also shown in Figure 4. The e^+LiX dissociation energies for the $e^+LiX \rightarrow Li^+ + PsX$ disassociation channel were also computed. A summary of these results, along with the experimental bond lengths and vibrational energies for the LiX and e^+LiX systems, are given in Table 9.

At 1.6856 Å, the relaxed e^+LiH geometry is approximately 4.8% longer than the optimized LiH geometry of 1.6088 Å. In comparison, Buenker *et al.* performed multireference (MR) single- and double-excitation configuration interaction calculations on LiH and e^+LiH and obtained a bond relaxation of 10.1%. Adding the MR-Davidson correction, they obtained a bond relaxation of 11.4%, [8] which was closer to the equilibrium bond length of 1.772 Å reported by Strasburger using the explicitly correlated Gaussian method. [58]

The equilibrium bond length for e^+LiF is 1.6556 Å or 2.6% longer than the calculated value for LiF of 1.6143 Å. Mella *et al.* computed the energy and geometry of e^+LiF close to equilibrium by means of diffusion Monte Carlo (DMC) simulations and obtained a e^+LiF equilibrium bond length of 1.68 Å, [35] which agrees well with the NEO-MP2 results. Further improvement of the electron-positron correlation in the calculation will result in more relaxation and better agreement. Also, the NEO-MP2 e^+LiF dissociation energy for the $e^+LiF \rightarrow Li^+ + PsF$ channel is 1.588220 eV, compared to the DMC value of 2.10 eV.

Using Equation (61) and the vibrational energy levels computed from the Morse potential fits to the NEO-MP2 PECs in Figure 4, the positron kinetic energies which will result in VFRs for the three systems were determined. These are listed in Table 10 by the corresponding LiX and e^+LiX vibrational energy levels. From these results, we see that for LiH and LiF, VFRs involving the ground vibrational state of the nonpositronic molecule ($\upsilon = 0$) are resonances with vibrational modes of $\upsilon = 3'$ of the positronic systems. For LiCl, the

Table 8: X^-, Li^-, PsX, LiPs, and e^+LiX (X = H, F, and Cl) even-tempered basis set parameters and energies optimized at the MP2 and NEO-MP2 levels. Calculations for systems containing Li, F, and Cl were carried out with an electronic frozen core.

Basis	α_s	β_s	α_p	β_p	α_d	β_d	E_{MP2}
H^- e^- 9s5p4d	0.004082	3.12031	0.018186	2.74250	0.033809	2.77814	−0.516892
PsH e^- 9s5p4d	0.018298	2.66531	0.029626	2.46850	0.041616	2.66534	−0.731866
e^+ 6s4p4d	0.012420	3.91161	0.029026	2.36631	0.022201	3.55911	
Li^- e^- 14s4p4d	0.002204	2.77725	0.006075	2.33755	0.010639	2.13914	−7.445067
LiPs e^- 14s4p4d	0.010922	2.59721	0.015232	2.16030	0.023429	1.91744	−7.591488
e^+ 5s4p4d	0.007060	3.89961	0.015801	2.03951	0.011862	3.87811	
F^- e^- 19s9p5d	0.045316	2.25429	0.024415	2.64280	0.102926	2.45140	−99.739172
PsF e^- 19s9p5d	0.032866	2.36649	0.033128	2.56460	0.075036	2.58720	−99.940104
e^+ 5s3p2d	0.010759	2.89690	0.032968	2.92950	0.061986	2.90720	
Cl^- e^- 24s17p8d	0.108252	2.10290	0.017951	2.05620	0.051060	2.01310	−459.756831
PsCl e^- 24s17p8d	0.111602	2.09090	0.024590	2.01730	0.054670	2.00370	−459.926481
e^+ 5s5p1d	0.003760	9.11710	0.025740	1.98550	0.050800	6.16980	
e^+LiH [a]							−8.029122
Li e^+ 5s4p4d	0.071319	2.76127	0.006019	3.72273	0.017703	5.58951	
H e^+ 6s4p4d	0.007813	1.80596	0.004562	3.66884	0.017550	3.46504	
e^+LiF [a]							−107.273739
Li e^+ 5s4p4d	0.035665	4.62001	0.002043	2.37254	0.007033	2.05041	
F e^+ 5s3p2d	0.000679	2.62154	0.022364	3.32900	0.059158	3.04560	
e^+LiCl [a]							−467.244967
Li e^+ 5s4p4d	0.026871	1.98041	0.029943	1.50001	0.004112	2.41161	
Cl e^+ 5s5p1d	0.061890	2.16070	0.002380	3.63710	0.026260	3.88720	

[a] LiPs and PsX e^- basis sets were held fixed during the e^+ even-tempered parameter and geometry optimizations for the e^+LiX systems.

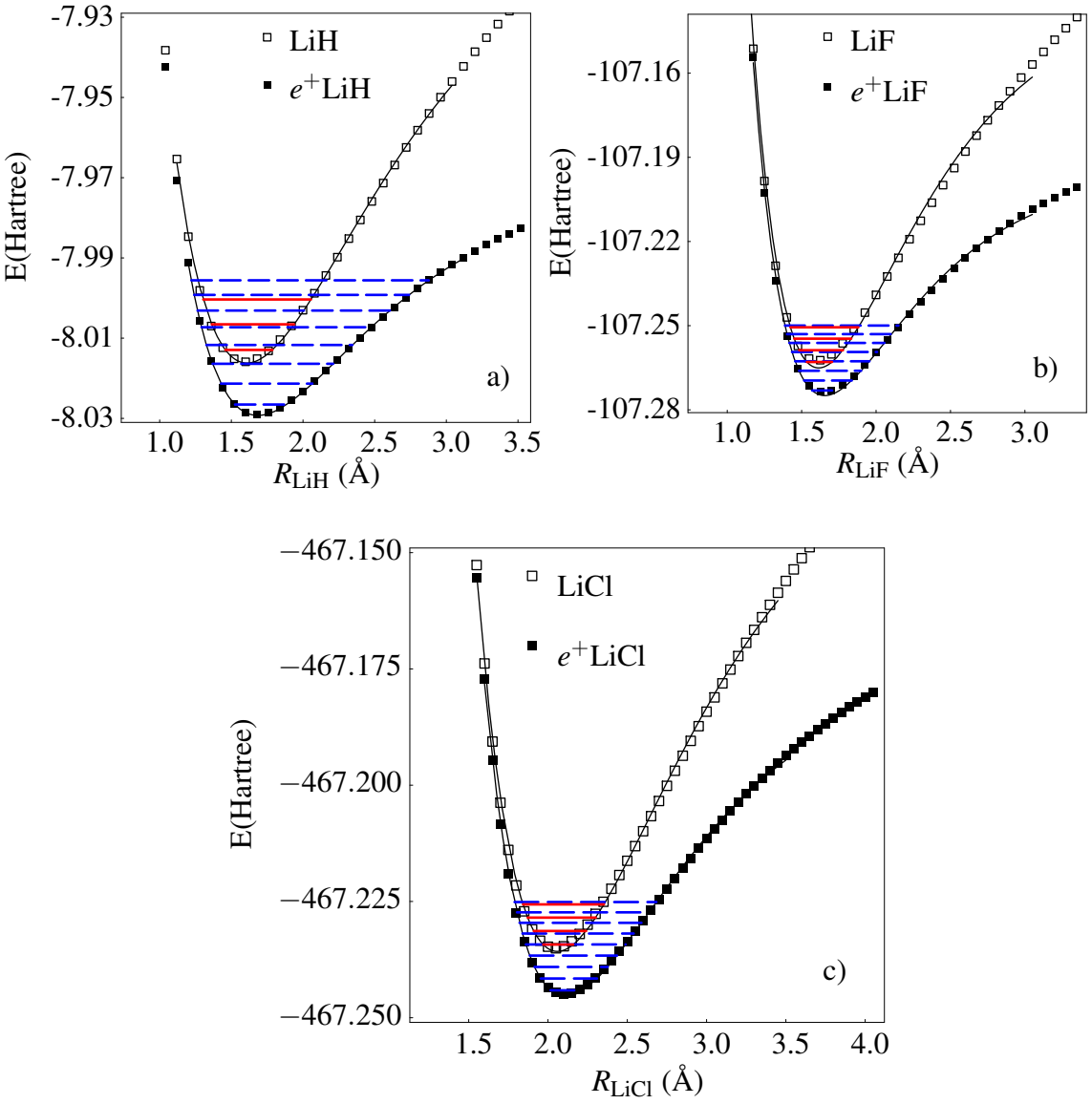

Figure 4: LiX and e^+LiX (X = H, F, Cl) MP2 and NEO-MP2 potential energy curves, with a frozen electronic core on the Li, F, and Cl atoms and electronic and positronic basis sets from Table 8. The Morse potential fits used to compute the geometries and vibrational parameters are also shown. The vibrational energy levels of LiX (—) and e^+LiX (– –) used to compute the VFR in Table 10 are also shown.

Table 9: LiX and e^+LiX (X = H, F, and Cl) MP2 and NEO-MP2 results. Listed are the calculated bond lengths (R_e and R_0), experimental bond lengths (R_{exp}), dissociation energies (Δ), harmonic constants (ω_e), anharmonic constants ($\omega_e\chi_e$), and calculated and experimental vibrational energies (ν and ν_{exp}, respectively). A frozen electronic core on Li, F, and Cl was used, and electronic and positronic basis sets were taken from Table 8.

	R_e (Å)	R_0 (Å)	R_{exp} (Å)[a]	Δ (eV)[b]	ω_e (cm^{-1})	$\omega_e\chi_e$ (cm^{-1})	ν (cm^{-1})	ν_{exp} (cm^{-1})[a]
LiH	1.6007	1.6088	1.5957	-	1441	20.3	1400	1406
e^+LiH	1.6810	1.6856	-	0.545365	1201	28.8	1144	-
LiF	1.6003	1.6143	1.5639	-	934	7.5	919	910
e^+LiF	1.6476	1.6556	-	1.588220	799	9.2	781	-
LiCl	2.0455	2.0592	2.0207	-	648	4.2	640	643
e^+LiCl	2.1029	2.1111	-	1.169670	566	5.1	556	-

[a]Experimental bond lengths and vibrational energies are from the CCCBDB. [9]

[b]The e^+LiX dissociation energies are computed for the $e^+\text{LiX} \rightarrow \text{Li}^+ + \text{PsX}$ disassociation channel, with an Li^+ MP2 energy of -7.274638 Hartree and zero-point energy corrections for e^+LiX. PsX MP2 energies are taken from Table 8.

$\upsilon = 0$ level will resonate with modes of $\upsilon = 4'$ or higher. For all three of the molecules, if the nonpositronic system is in the $\upsilon = 1$ state, two additional vibrational energy levels become inaccesible by VFR.

Clearly, anharmonic effects play a significant role in calculating the vibrational energy levels of the e^+LiX systems. Upon addition of the positron, the magnitude of ω_e decreases significantly, while $\omega_e\chi_e$ increases. This has the net effect of reducing the vibrational energy of the positronic system by a greater amount than if a simple harmonic approximation is used for the PEC fit. The calculated VFRs are also affected, since the spacing decreases more rapidly in the positronic system for higher vibrational energy levels.

For each of the systems, the decrease in the stiffness of the bonds upon addition of the positron is expected and correlates with the increase in R_e. Attachment of the positron has less effect on the vibrational spectra of the LiX systems going from H to F to Cl. The calculated e^+LiF vibrational energy is 781 cm^{-1}, a decrease of 15.0% from the LiF value of 919 cm^{-1}. This decrease is less than the 18.3% drop in the vibrational energy of LiH (from

Table 10: LiX (X = H, F, and Cl) VFRs computed from MP2 and NEO-MP2 PECs. The positron kinetic energy (ε_k) associated with each VFR is listed by the corresponding LiX and e^+LiX vibrational energy levels ($\upsilon = n$ and $\upsilon = n'$, respectively).

LiH			LiF			LiCl		
n	n'	ε_k (eV)	n	n'	ε_k (eV)	n	n'	ε_k (eV)
0	$3'$	0.033494	0	$3'$	0.006866	0	$4'$	0.000566
	$4'$	0.153845		$4'$	0.096847		$5'$	0.064482
	$5'$	0.267052		$5'$	0.184559		$6'$	0.127136
	$6'$	0.373115		$6'$	0.270001		$7'$	0.188529
	$7'$	0.472034		$7'$	0.353175		$8'$	0.248661
1	$5'$	0.093449	1	$5'$	0.070591	1	$6'$	0.047823
	$6'$	0.199512		$6'$	0.156034		$7'$	0.109216
	$7'$	0.298431		$7'$	0.239207		$8'$	0.169347
2	$6'$	0.030943	2	$6'$	0.043928	2	$7'$	0.030938
	$7'$	0.129862		$7'$	0.127101		$8'$	0.091069
			3	$7'$	0.016856	3	$8'$	0.013826

1400 cm^{-1} to 1144 cm^{-1}), and it is more than the 13.0% drop for LiCl (from 640 cm^{-1} to 556 cm^{-1}).

The trend in the shift of the vibrational energy is due to the nature of the positron MO on the molecule and can be understood by examining the positronic densities of the e^+LiX systems. The positronic densities for e^+LiX and the electronic densities for LiX are given in Figure 5. For the systems, as we go from X=H to F to Cl, it is clear that the positronic density is more localized on the X atom, and less positronic density is present in the interatomic spacing of the molecule. The trend explains why the presence of the positron has less effect on the vibrational energy of e^+LiF and e^+LiCl as compared to e^+LiH. The positron weakens the LiH bond more than the LiF and LiCl bonds.

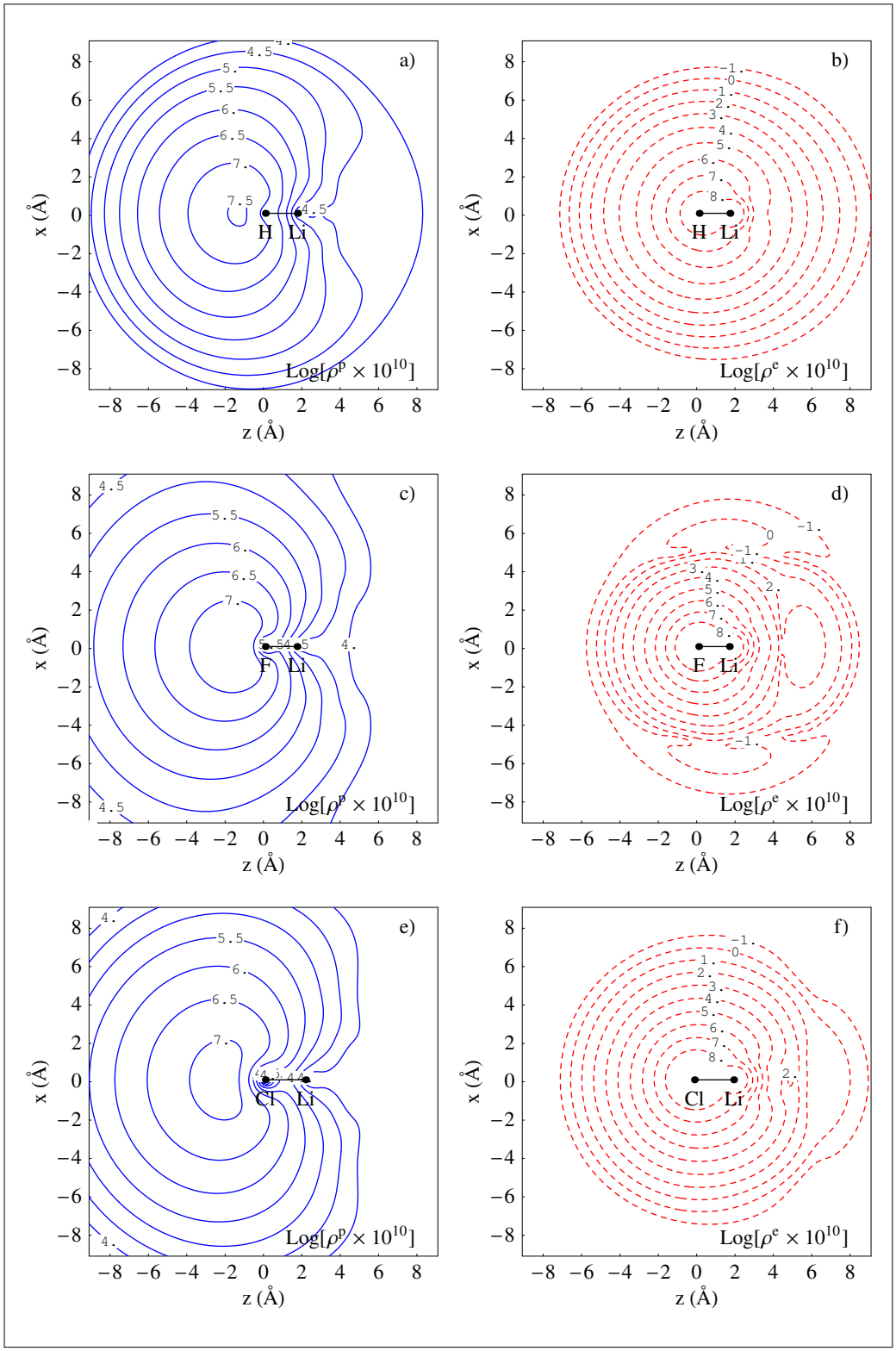

Figure 5: e^+LiX positronic density (—) and LiX electronic density (– –). [X = H (plots a and b), F (plots c and d), Cl (plots e and f)].

V. Positron Interaction With $B_{12}H_{12}^{2-}$

5.1 Introduction

In order to investigate the interaction of positrons with the $B_{12}H_{12}^{2-}$ dianion, we first need to understand the electronic structure of the molecule. This will guide the placement and choice of positron basis functions. While there have been several theoretical studies of icosahedral boranes in molecular and condensed phase form, to my knowledge, none have considered positron binding. For example, Green *et al.* performed RHF calculations at the double-zeta level on $B_{12}H_{12}^{2-}$, $CB_{11}H_{12}^{-}$, and 1,12-$C_2B_{10}H_{12}$. [20] These calculations did not include correlation effects, which we have seen are critical to describing positron interactions.

As in the previous chapter, electron-electron and electron-positron correlation are included with the MP2 method. [40] All calculations were performed with GAMESS, [49] and the positron is included through the use of the NEO method, [60,71] modified to handle positrons as discussed in Chapter III.

The $B_{12}H_{12}^{2-}$ molecule has very high symmetry (I_h point group), but the MP2 method as it is implemented in NEO only supports a certain number of Abelian point groups. The largest Abelian subgroup of I_h that is supported by NEO-MP2 is C_{2v}, and all of the calculations in this chapter were performed with this symmetry. Also, the cc-pVDZ electronic basis set is used. [15]

5.2 Computational Results

The MP2-optimized geometry of $B_{12}H_{12}^{2-}$, is shown in Figure 6, and the coordinates of the C_{2v} symmetry-unique atoms are given in Table 11.[1] The minimized MP2 energy is -304.544568, with a HF energy of -303.302892 and a second-order correction of -1.241676. The optimized bond lengths are 1.8042 Å and 1.2138 Å for R_{BB} and R_{BH}, respectively. The computed value of R_{BH} agrees very well with the experimental value of Kozlova *et al.* of 1.21±0.01 Å obtained with nuclear magnetic resonance (NMR). [30] Tir-

[1] Figures 6, 7, 8, and 9 were generated with MacMolPlt. [7]

Table 11: MP2-optimized C_{2v} symmetry-unique atomic coordinates of $B_{12}H_{12}^{2-}$. The complete molecule is shown in Figure 6.

Atom	x	y	z
B1	1.459608	0.902091	0.000000
B2	0.000000	1.459609	0.902091
B3	0.000000	1.459609	−0.902091
B4	0.902091	0.000000	1.459608
B5	0.902091	0.000000	−1.459608
H6	2.492097	1.540205	0.000000
H7	0.000000	2.492097	1.540205
H8	0.000000	2.492097	−1.540205
H9	1.540205	0.000000	2.492097
H10	1.540205	0.000000	−2.492097

itiris also obtained NMR values for R_{BB} and R_{BH} of 1.20 Å and 1.78 Å, respectively, [67], which agree well with these results. In comparison, the minimum RHF energy of Green *et al.* with the smaller 4-31G* electronic basis set was −302.963784, and the R_{BB} and R_{BH} bond lengths were 1.7922 Å and 1.2015 Å, respectively. The $B_{12}H_{12}^{2-}$ molecular geometry was also optimized at the RHF level with the cc-pVDZ basis, obtaining a minimum energy of −303.302977 Hartree and R_{BB} and R_{BH} bond lengths of 1.8026 Å and 1.2102 Å, respectively. From these results, we see that both the larger basis set and the inclusion of electron-electron correlation at the MP2 level have the effect of increasing the bond lengths.

In order to determine the nature of the electronic wavefunction of the $B_{12}H_{12}^{2-}$ molecule, plots of the MP2 molecular electrostatic potential (MESP) are given in Figures 7 and 8. The plots show both the negative and positive contours in increments of 0.025 a.u. with a maximum contour value of 1.0 a.u. There are five regions of interest (ROIs) for the MESP: (1) inside the boron cage; (2) in the vicinity of the boron cage; (3) between the boron cage and the hydrogen cage; (4) in the vicinity of the hydrogen cage; and (5) outside of the hydrogen cage.

Figure 7 is a two-dimensional slice through the MESP in an "equatorial plane" which bisects ten boron-boron bonds perpendicular to a C_5 rotation axis. In ROI 1, the MESP is negative with a value of approximately −0.15 a.u. at the center of the cage. From the center

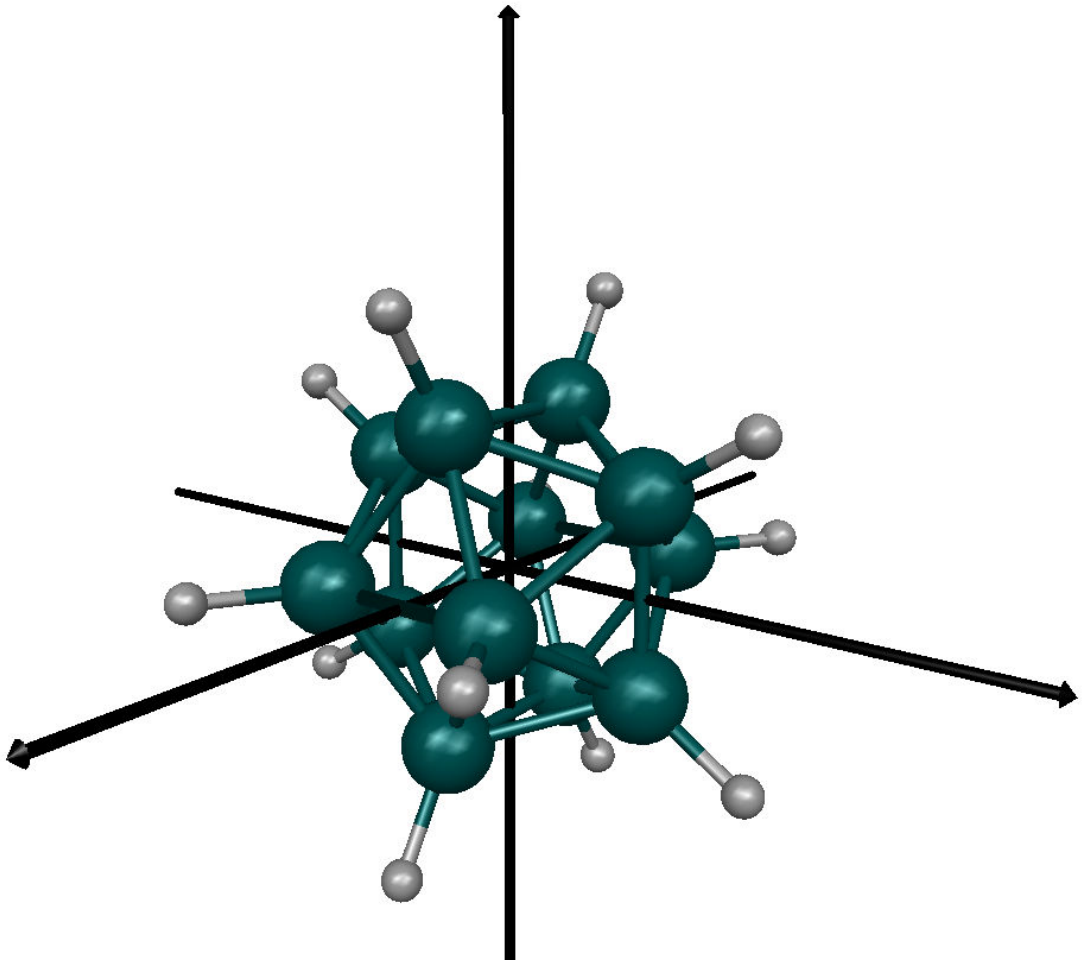

Figure 6: $B_{12}H_{12}^{2-}$ C_{2v} geometry. The three Cartesian planes (*xy*-, *xz*-, and *yz*-plane) each have four boron atoms and four hydrogen atoms. The symmetry-unique atomic coordinates are given in Table 11.

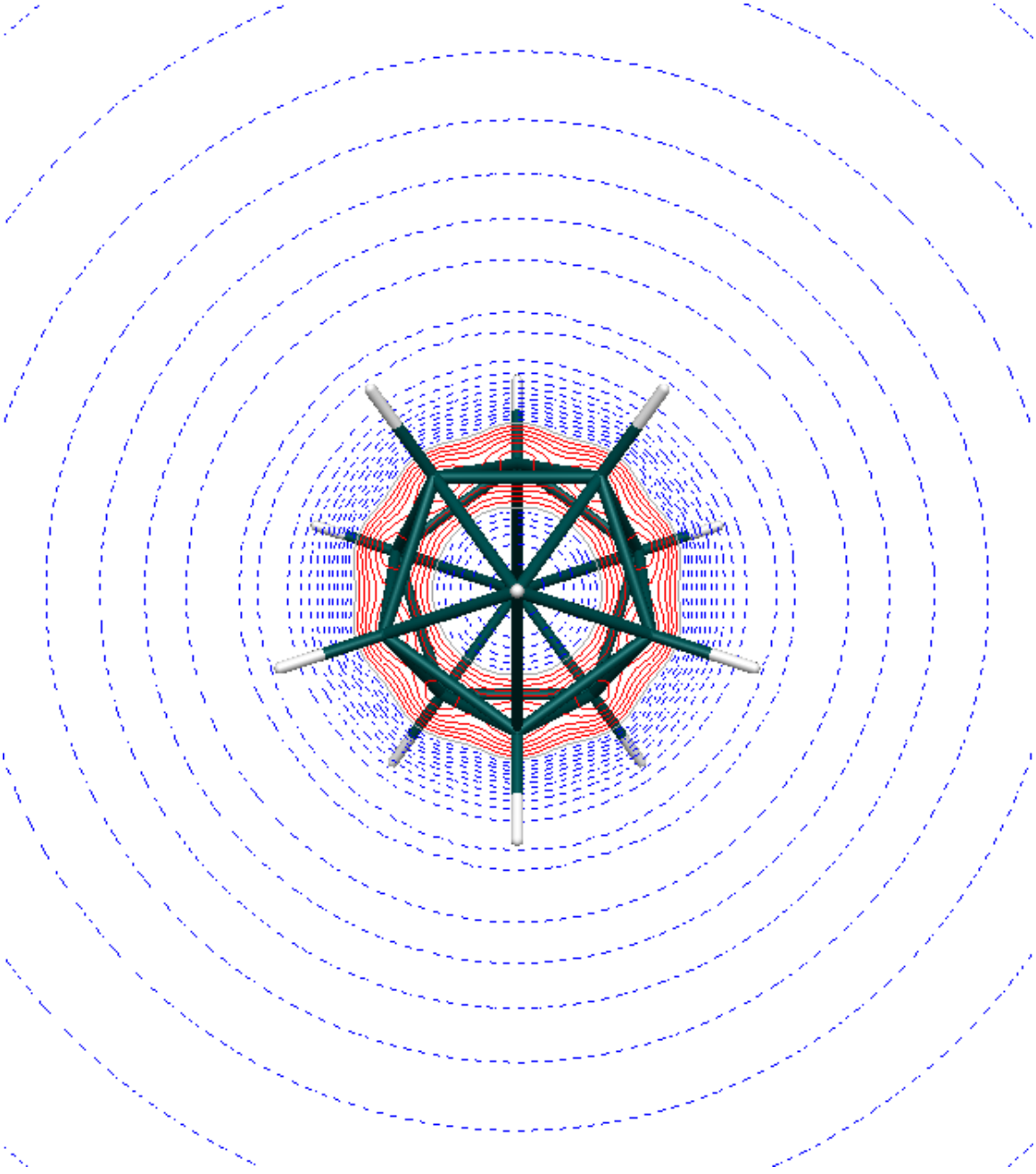

Figure 7: $B_{12}H_{12}^{2-}$ two-dimensional molecular electrostatic potential (MESP) in equatorial plane of C_{2v} geometry. The positive contours (—) and negative contours (– –) are shown in increments of 0.025 a.u. with a maximum value of 1.0 a.u. This cut of the MESP shows very little divergence from spherical symmetry.

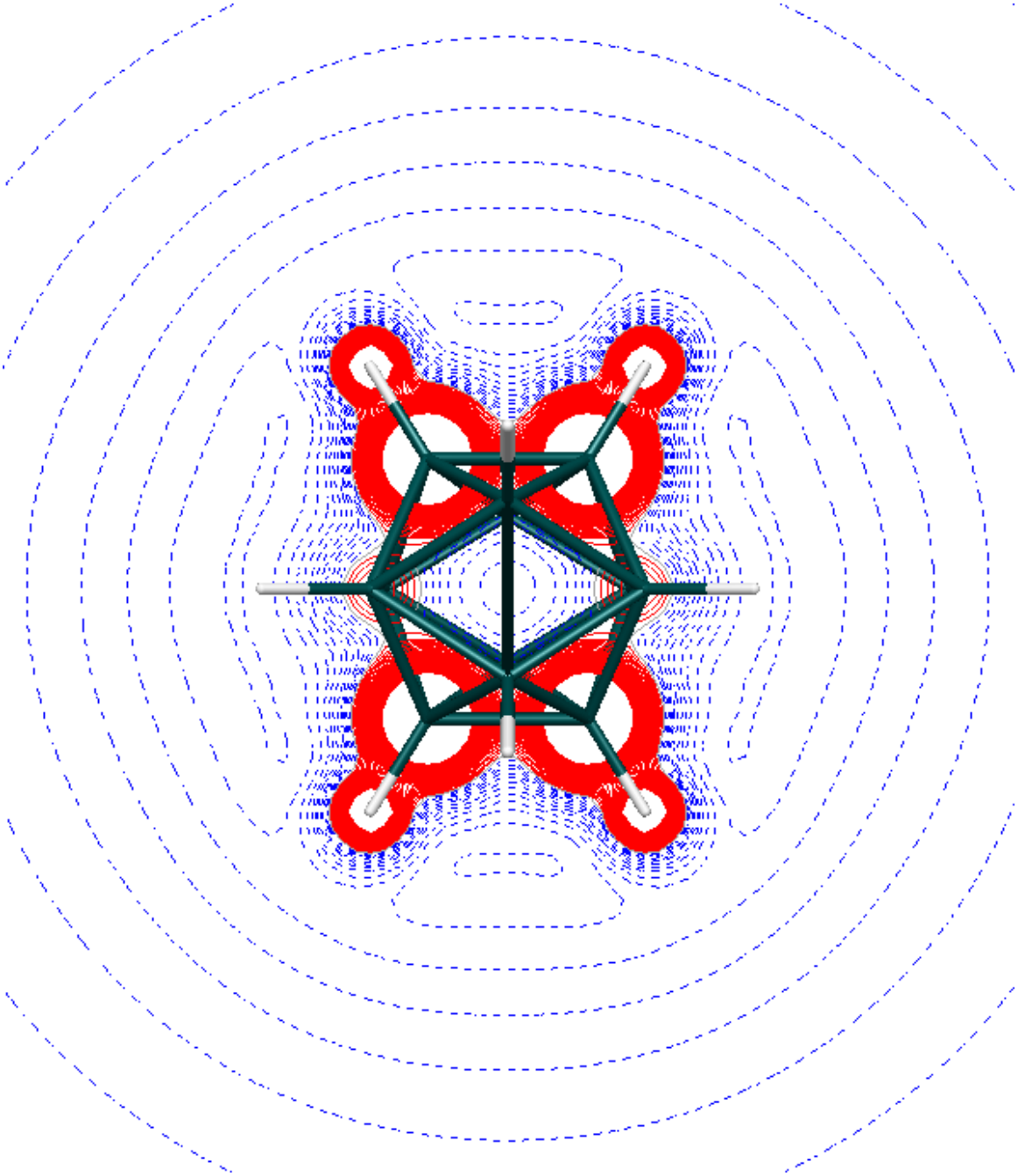

Figure 8: $B_{12}H_{12}^{2-}$ two-dimensional molecular electrostatic potential (MESP) in the Cartesian plane of the C_{2v} geometry of Figure 6. The positive contours (—) and negative contours (– –) are shown in increments of 0.025 a.u. with a maximum value of 1.0 a.u. This cut of the MESP shows significant deviation from spherical symmetry in the vicinity of the boron and hydrogen atoms. Away from the atoms, the MESP retains spherical symmetry.

of the cage to the edge of ROI 2, the magnitude of the MESP decreases gradually to zero, and the MESP is positive in ROI 2. The plot reveals very little deviation from spherical symmetry, most of which is confined to ROI 2, at the triangular faces of the boron cage. In ROI 3, between the boron and hydrogen cages, the MESP goes from positive to negative, and remains negative in ROI 4, with a peak in the magnitude in this region. The magnitude of the MESP decreases gradually outside of the hydrogen cage, in ROI 5.

Figure 8 shows a two-dimensional slice through the MESP in one of the Cartesian planes (the *xy*-, *xz*-, and *yz*-planes are equivalent). This plot is in stark contrast to the plot in Figure 7, since it cuts through four boron atoms and four hydrogen atoms to reveal drastic deviation from spherical symmetry. This deviation is confined to the areas immediately surrounding each of the atoms and the two bisected boron-boron bonds in ROIs 2, 3, and 4. In ROI 4, the MESP is still negative in the areas between the hydrogen nuclei, and there is a peak in the magnitude at the radius of the hydrogen cage.

The best positronic basis set for $e^+B_{12}H_{12}^{2-}$ would include at least as many s-, p-, and d-type Gaussian primitives at each of the boron and hydrogen nuclei as are included in the electronic basis set. Unfortunately, convergence of the self-consistent field (SCF) method is problematic for this system, and a systematic approach to adding positronic basis functions to the molecule is required. The analysis of the MESP for $B_{12}H_{12}^{2-}$ suggests that a good initial positronic basis set for $e^+B_{12}H_{12}^{2-}$ is a set of s-type Gaussian primitives located at the center of the cage. For an initial positronic basis set of this type, a set of twelve even-tempered functions were chosen in which the ith exponent is given by $\zeta_i = \alpha\beta^i$. Based on previous experience developing even-tempered positronic basis sets, an α of 0.0001 and a β of 3.0 was used for this initial basis.

Figure 9 shows the NEO-HF positronic density for $e^+B_{12}H_{12}^{2-}$ with the cc-pVDZ electronic basis set on the atoms and the even-tempered [12s] positronic basis set at the center of the cage. This calculation was performed at the MP2-optimized geometry of $B_{12}H_{12}^{2-}$. It is clear from the plot that the positronic density is very diffuse, especially considering the maximum contour has a value of 0.001 a.u. and the contours are shown at every 0.00005 a.u.

Table 12: MP2 and NEO-MP2 results for $B_{12}H_{12}^{2-}$ and $e^+B_{12}H_{12}^{2-}$. The HF and NEO-HF energies (E_{HF}), MP2 and NEO-MP2 energies (E_{MP2}), and second-order electron-electron and electron-positron energies (E_{MP2}^{ee} and E_{MP2}^{ep}, respectively) are listed for each system. Also listed is the change in each parameter with the addition of the positron. Energies are in Hartrees.

	$B_{12}H_{12}^{2-}$	$e^+B_{12}H_{12}^{2-}$	Change
E_{HF}	−303.302892	−303.488281	−0.185389
E_{MP2}^{ee}	−1.241676	−1.241423	0.000253
E_{MP2}^{ep}	-	−0.000619	−0.000619
E_{MP2}	−304.544568	−304.730322	−0.185755

As expected, there is a peak in the positron density just outside of the hydrogen cage, in the same location as the negative peak in the MP2 MESP of Figures 7 and 8. Surprisingly, there is very little positronic density in the center of the cage at the HF level. This is perhaps due to the lack of electron-positron correlation in the NEO-HF calculation.

Although a plot of the MP2 positronic density is unavailable, a NEO-MP2 energy of −304.730322 was obtained for the $e^+B_{12}H_{12}^{2-}$ system. The NEO-HF energy of the system is −303.488281, and the electron-electron and electron-positron correlation energies are −1.241423 and −0.000619, respectively. The results of the MP2 and NEO-MP2 calculations are summarized in Table 12.

As discussed in Section 2.2.4, knowledge of the IP for a system can be of help when predicting the qualitative nature of the corresponding positronic system. With the cc-pVDZ electronic basis set, the energy of the $B_{12}H_{12}^-$ was calculated to be -304.506336 Hartree, for an IP of 1.040335 eV. This value is much less than 6.8 eV, which indicates that the system should resemble $Ps[B_{12}H_{12}^-]$ instead of $e^+B_{12}H_{12}^{2-}$. As a consequence, a significant portion of the positronic density should be present at the center of the cage molecule.

5.3 Experimental Results

A PALS system was assembled and used to measure the positron lifetime of potassium dodecahydrododecaborate methanolate, $K_2B_{12}H_{12}\cdot CH_3OH$, which was obtained in

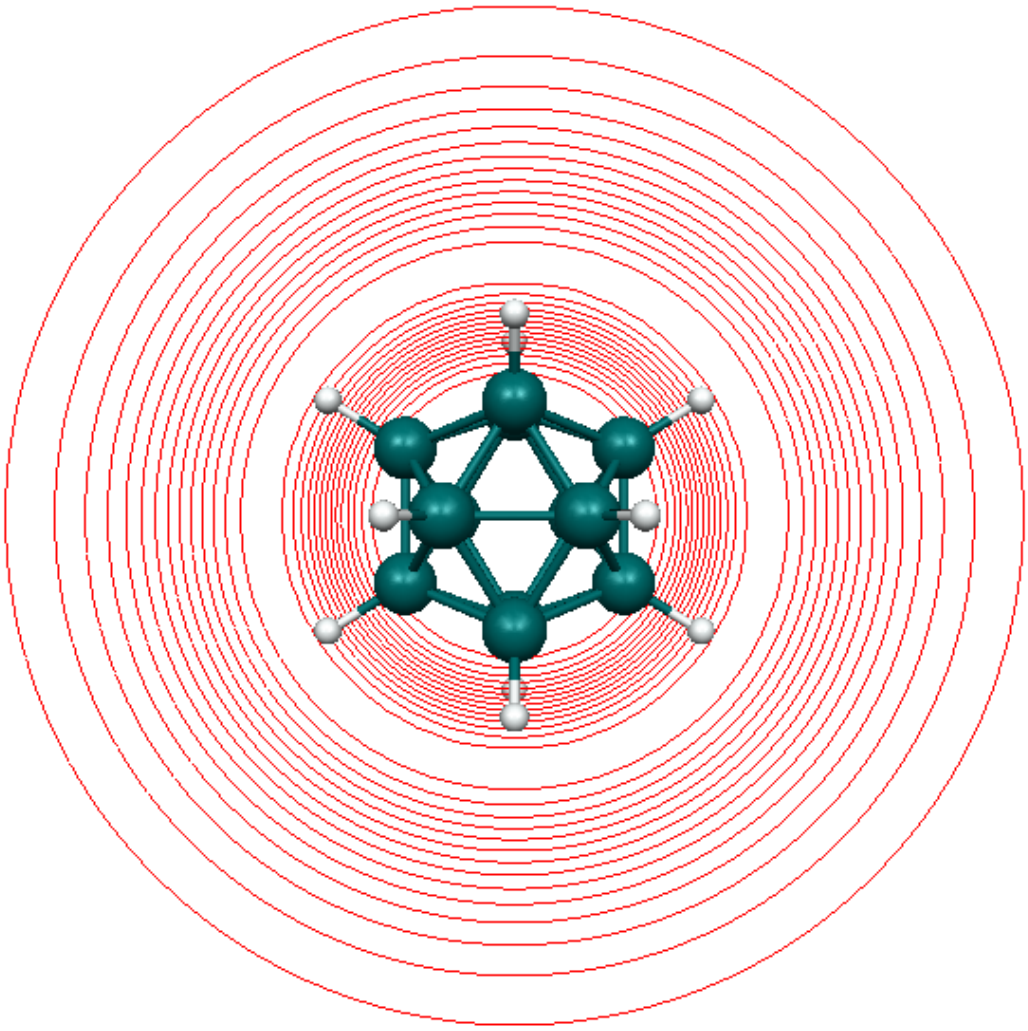

Figure 9: $e^{+}B_{12}H_{12}^{2-}$ NEO-HF positronic density. The two-dimensional positronic density (—) is shown. The maximum contour of the positronic density is 0.001 a.u. and there is 0.00005 a.u. between each contour. The basis sets consist of cc-pVDZ electronic basis functions on the B and H atoms and an even-tempered [12s] positronic basis set at the center of the cage.

powder form from BASF Corporation. The material is specified to be no more than 15.0% methanol. Suitable samples for PALS were made by pressing at 23,000 lb.

5.3.1 Positron Annihilation Lifetime Spectrometer. The schematic of the PALS system is shown in Figure 10. It is of the "fast-slow" type, in which timing pulses from the detectors are used to provide start and stop signals for an Ortec 565 time-to-amplitude converter (TAC) in the "fast" side, and energy pulses are used to provide a gate for a Canberra 9635 analog-to-digital converter (ADC) on the "slow" side. At the heart of the system is a pair of BaF_2 scintillation detectors with XP2020Q photomultiplier tubes (PMTs), which provide excellent energy resolution and timing characteristics.

The dynode signals from the PMTs, which provide the best energy information, are each sent to an Ortec 113 preamplifier and then amplified and shaped in an Ortec 672 spectroscopy amplifier. The unipolar outputs from the spectroscopy amplifiers are then sent to a pair of Ortec 551 timing/single channel analyzers (T/SCAs) to pick off valid start and stop signals (1.274 MeV and 0.511 MeV photons, respectively). The start and stop logic pulses are delayed in Ortec 416A gate and delay generators (G&DG) in order to compensate for the inherent delay of the TAC and ADC in the timing chain. The Ortec CO4020 quad logic module provides the gate signal to the ADC so that output from the TAC is processed only when both valid start and valid stop pulses occur in the specified time window.

The anode signals from the detectors, which provide the best timing information, are each sent directly to one of the channels of the Ortec 935 quad constant fraction discriminator (CFD), which provides the start and stop pulses for the TAC. The TAC output is then processed by the ADC and then the ethernet multichannel analyzer. The PALS spectrum is collected on a personal computer (PC) running Genie 2000. Analysis of the spectrum is accomplished with the PALS*fit* program. [42]

5.3.2 Positron Source and Sample Configuration. The positron source was prepared by depositing ^{22}NaCl between two 4 μm thick Mylar films and sealing with double-sided tape. "Carrier-free" ^{22}NaCl in aqueous solution was obtained from Isotope Products

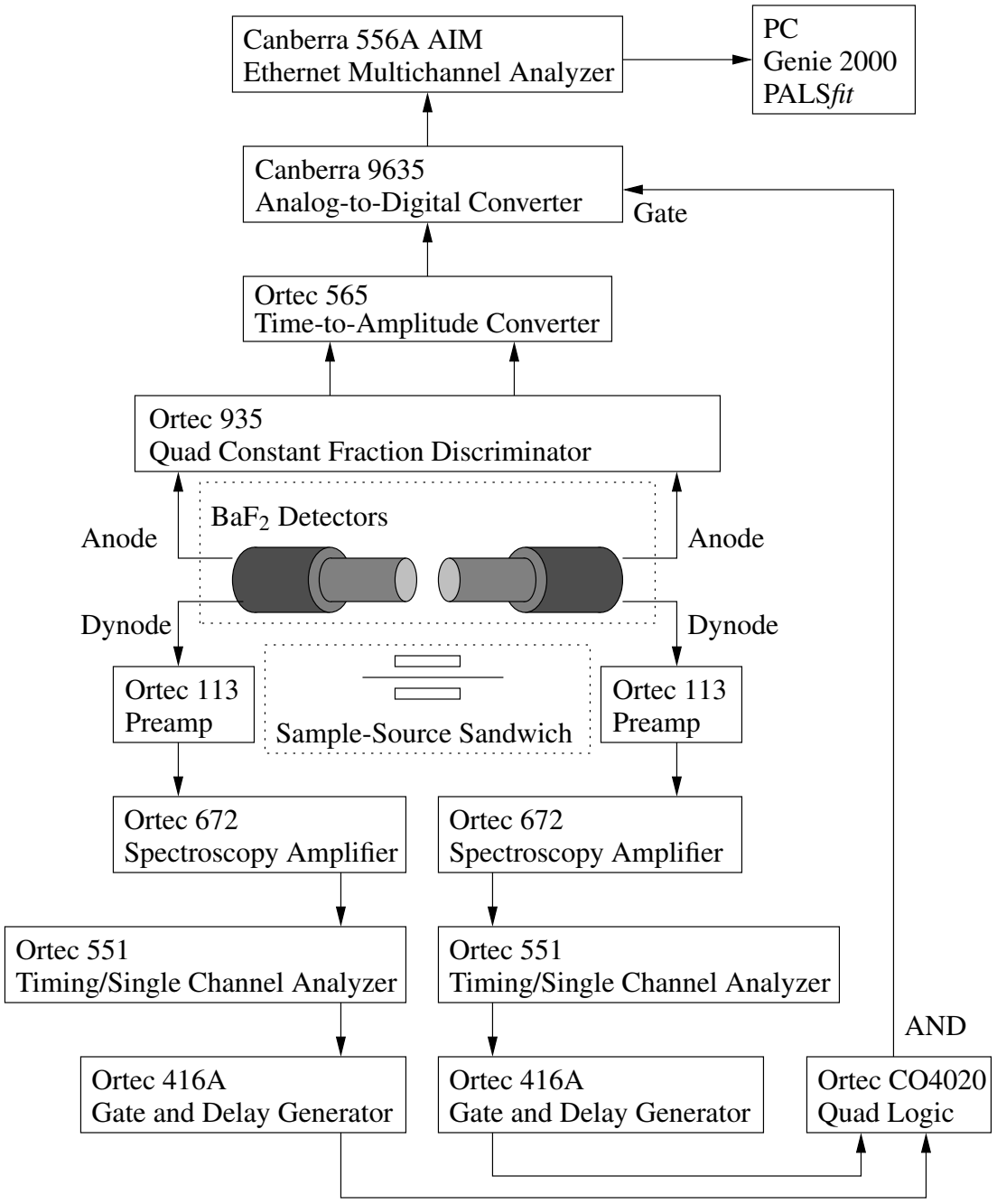

Figure 10: Schematic of PALS system.

for the source. In order to reduce the volume to be transferred to the Mylar film, 20 μCi of activity was transferred to a Teflon dish for drying under an infrared lamp. Unfortunately, the ^{22}NaCl solution contained an unexpected carrier, perhaps an organic from the extraction process, at a substantial concentration. This made it extremely difficult to transfer the desired activity to the Mylar film, and the final activity of our positron source was ≈ 7 μCi. The carrier also contaminated our positron source, causing a large source correction in our PALS spectra.

To ensure that most of the positrons from the source annihilated in the sample, a "source-sample sandwich" configuration was used. In this type of configuration, two identical samples are placed on either side of the positron source. The "source-sample sandwich" is placed next to the two detectors, which are placed end-to-end, as shown in Figure 10. This geometry limits the chances that both a 1.274 MeV 'birth' gamma and a 0.511 MeV annihilation photon will enter the same detector.

5.3.3 Data Analysis. The most difficult part of PALS is extracting physically meaningful parameters from measured spectra. Fortunately, the PALS*fit* program, [42] which was recently released in an updated form, provides a systematic method to go about this task. It is based on software that has been used extensively by the positron annihilation spectroscopy community. [26, 27] PALS*fit* consists of two distinct subprograms:

- *ResolutionFit* determines the time resolution function of the PALS system
- *PositronFit* extracts lifetimes and intensities from lifetime spectra

With ResolutionFit, the parameters which determine the shape of the resolution function of the spectrometer are fitted by analysis of lifetime spectra which contain mainly one component. The extracted resolution function is then used in the PositronFit subprogram for analyzing more complicated spectra.

A PALS spectrum of single crystal tungsten, shown in Figure 11, was used in conjunction with ResolutionFit to determine the resolution function of the PALS system. The bottom plot in Figure 11 shows the spectrum and the fit obtained with ResolutionFit, and

the top plot is the residual (the data points minus the fit). The resolution function consists of three Gaussian functions with exponents of 2.180 ns, 1.220 ns, and 1.515 ns; shifts of 0.00 ns, -0.61 ns, and -0.66 ns; and intensities of 80%, 10%, and 10%, respectively. During the optimization of the resolution function, the known lifetime of tungsten was fixed at 0.1200 ns, and additional lifetimes of 12.8446 ± 0.2089 ns and 1.8882 ± 0.0136 ns were found. The intensities of the tungsten lifetime and the additional lifetimes were 33.8 ± 0.2 %, 20.3 ± 0.1 %, and 45.9 ± 0.2 %, respectively. The χ^2 value for the fit was 3745.

5.3.4 $K_2B_{12}H_{12}\cdot CH_3OH$ Positron Lifetime. The positron lifetime spectrum for $K_2B_{12}H_{12}\cdot CH_3OH$, the fit obtained with PositronFit, and the associated residual plot is shown in Figure 12. From this analysis, a positron lifetime of 0.2645 ± 0.0077 ns is obtained for $K_2B_{12}H_{12}\cdot CH_3OH$ with an intensity of 25.9 ± 0.3 %. Again, two additional lifetimes were obtained, and they had similar values and intensities as found with the tungsten sample. The additional lifetimes were 12.6651 ± 0.1872 ns and 1.9450 ± 0.0154 ns, with intensities of 20.2 ± 0.1 % and 53.9 ± 0.3 %, respectively. Due to the presence of the two additional lifetimes in both spectra, it is believed that they are due to positrons annihilating in the source.

5.3.5 Effective Number of Electrons. Following the methodology described in Section 2.1.2.1, an effective number of electrons binding with the positron in $e^+B_{12}H_{12}^{2-}$ can be calculated from the measured annihilation rate ($\lambda = 1/\tau = 3.7807 \pm 0.1069 ns^{-1}$). The calculation yields an effective number of electrons of 1.89 or 0.94 of the two electrons per $B_{12}H_{12}^{2-}$ dianion. This result coincides with the conclusions of Section 5.2 that there must be significant positron density at the center of the molecular cage.

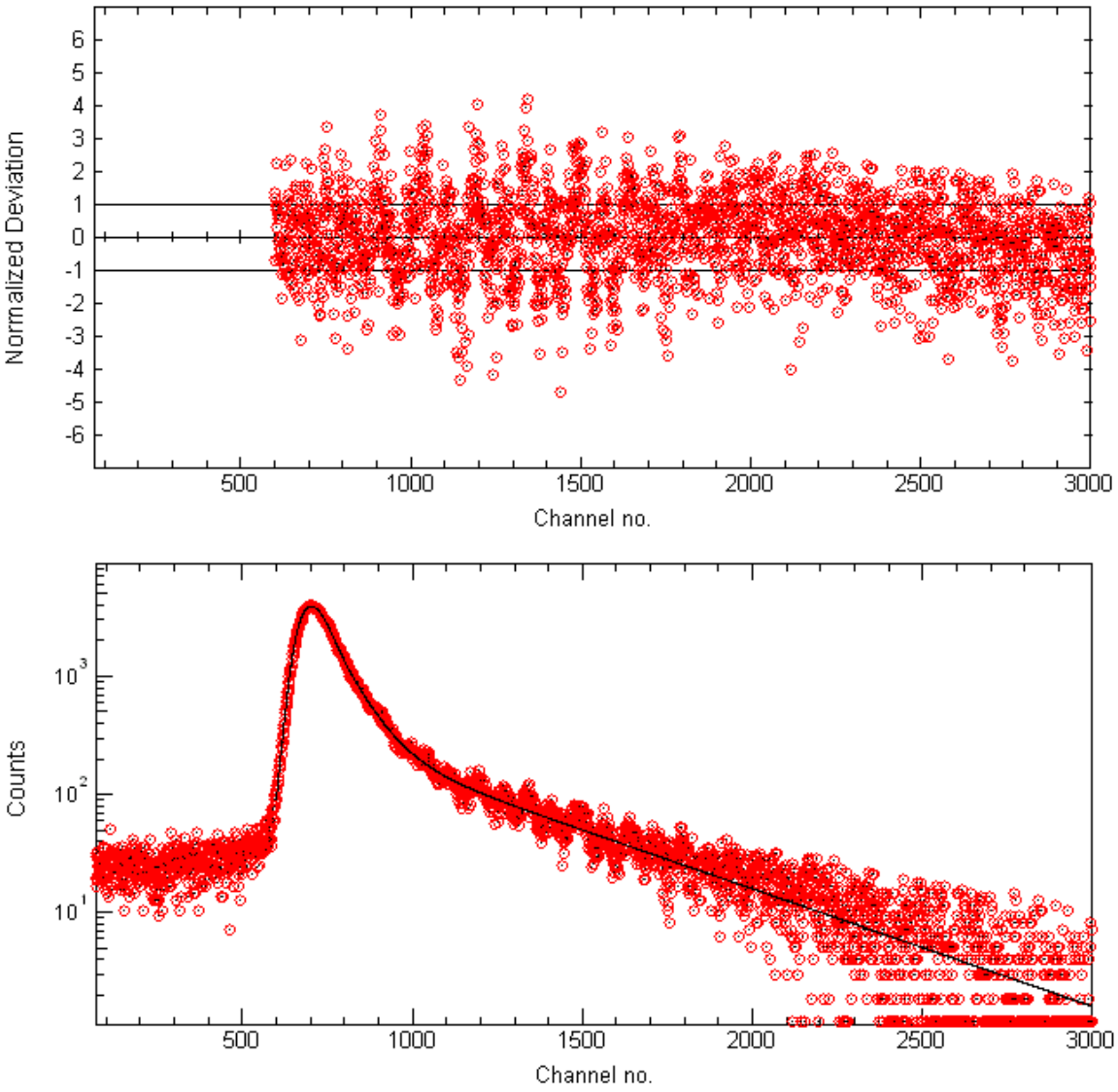

Figure 11: Tungsten positron lifetime spectrum used to determine the resolution function of the PALS system. The bottom plot shows the spectrum and the fit obtained with ResolutionFit. The top plot is the residual (the data points minus the fit).

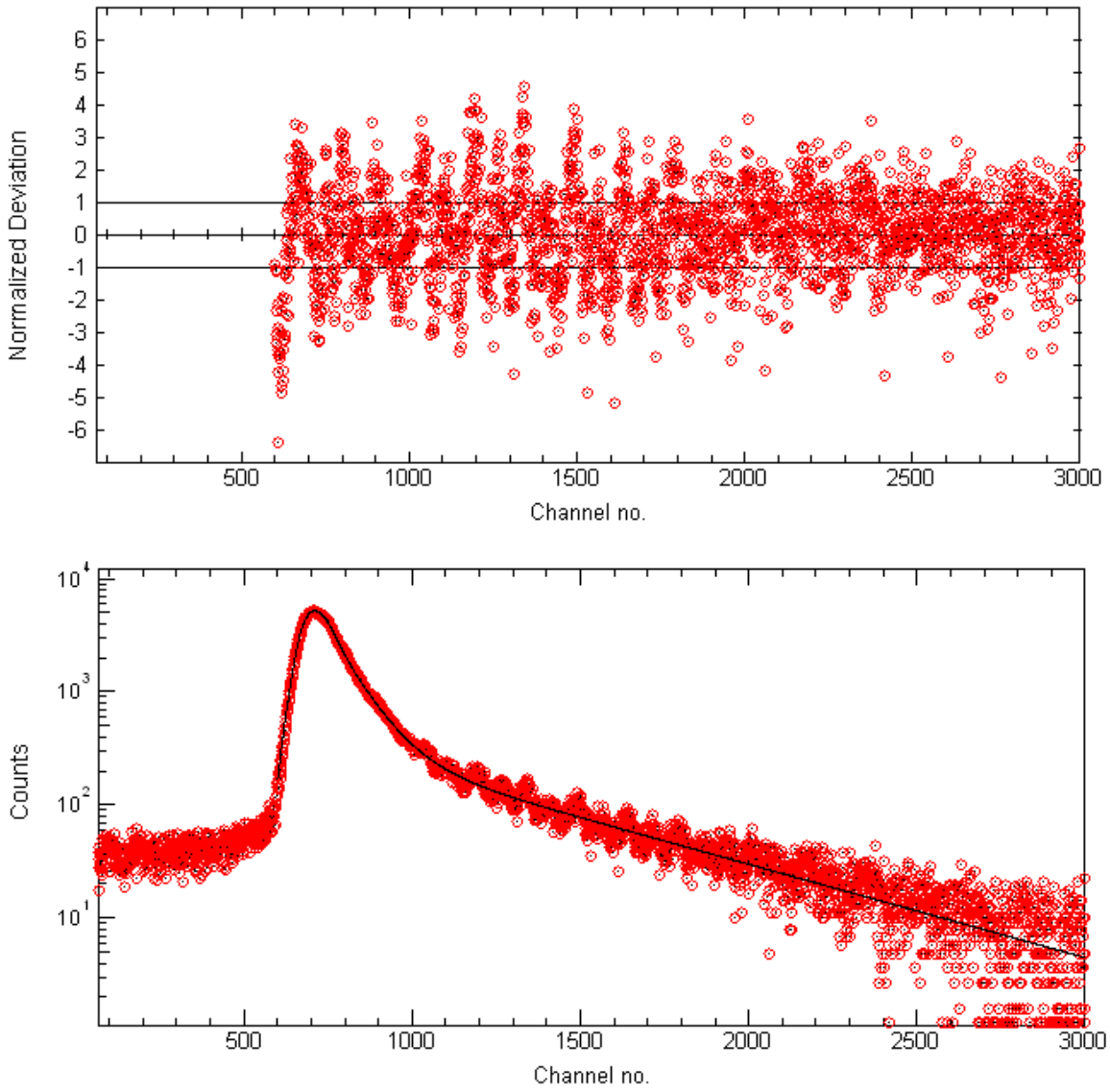

Figure 12: $K_2B_{12}H_{12}\cdot CH_3OH$ positron lifetime spectrum. The bottom plot shows the spectrum and the fit obtained with PositronFit. The top plot is the residual (the data points minus the fit.

VI. Summary and Concluding Remarks

6.1 Summary of Results

Modifications to the NEO method within the GAMESS program have allowed the calculation of wavefunctions of several positronic molecules. The methods made available include the HF, MP2, CI, CASSCF, and FCI methods. The NEO-HF method provides the energy corresponding to the single-configurational positronic-electronic wavefunction, minimized with respect to the MOs, which are expressed as linear combinations of Gaussian basis functions. The electron-electron and electron-positron correlation can be treated in the NEO framework with NEO-MP2 or multiconfigurational methods such as the NEO-CI and NEO-CASSCF. Also, the methodology for calculating positron-electron annihilation rates based on NEO-HF and NEO-MP2 wavefunctions was implemented.

The modified NEO method was applied to the PsH, e^+LiX (X = H, F, Cl), and $e^+B_{12}H_{12}^{2-}$ systems. Electronic and positronic basis sets were developed for these systems, and it was found that the optimization of "molecular" positronic basis sets was necessary in order to maximize the amount of electron-positron correlation energy recovered. Also, for the e^+LiH system, it was found that a frozen electronic core on the Li atom restricted the active space of electrons to exclude the 1s shell for Li, improving the balance in correlation energy between the LiH and e^+LiH systems.

Geometries and vibrational energy levels were computed for the LiX and e^+LiX systems at the MP2 and NEO-MP2 levels. It was found that addition of the positron causes the LiX bond to lengthen and become less stiff. Also, the anharmonic terms increase upon addition of the positron. The differences between the vibrational constants of the positronic and non-positronic systems have significant implications in the calculation of the VFRs for these systems.

The positron lifetime in pressed samples of $K_2B_{12}H_{12}\cdot CH_3OH$ was measured to be 0.2645 ± 0.0077 ns, and this result was interpreted with NEO calculations. An effective number of electrons of 1.88 was calculated from the measured annihilation rate, indicating that there is significant positron density both inside and outside of the $B_{12}H_{12}^{2-}$ cage. The

computational results do not currently agree with the experimental results due to the lack of sufficient electron-positron correlation in the calculation.

6.2 Future Work

This project has uncovered several areas upon which to focus future efforts. Some of these possibilities are presented here, and are categorized into three main areas: (1) code development; (2) experimental development; and (3) positronic systems for computational and experimental focus.

6.2.1 Code Development. Even for relatively small systems, a large number of positronic and electronic basis functions are required for an accurate description of the wavefunction. It is also clear that higher angular momentum (*e.g.*, f- and g-type) basis functions are necessary for capture of electron-positron correlation effects. It is recommended that the primary focus in code development be the implementation of higher angular momentum positronic basis functions and parallelization of the code.

Unfortunately, simply adding numerous higher angular momentum basis functions will not provide the amount of electron-positron correlation necessary. The next step is to develop multireference methods such as multi-reference perturbation theory (MRPT) and CASSCF for positrons. Of course, the current implementation of NEO includes a CASSCF method; however, it requires that all positron MOs be in the active space, a limitation that needs to be lifted in order to compute larger systems. Also, explicitly correlated Gaussian basis functions have recently been added to NEO, [62] and this method should be investigated for modeling positronic systems.

Another area of code development that requires immediate attention involves the way in which positron basis function centers are treated in NEO. Since NEO was developed to treat quantum nuclei, the positron basis function centers currently help define the symmetry of the system. In reality, the positronic wavefunction *has* the symmetry of the system. In other words, the positron basis function centers should be treated as a second set of electron basis function centers, not nuclear basis function centers. There is a distinct difference,

and a change will require a significant bifurcation of the NEO code. There are several advantages to changing the way in which positron basis function centers are treated in NEO. For example, currently NEO is unable to use certain symmetry groups which have inversion symmetry (*e.g.*, T_d). This limitation could be eliminated by treating the positron identically as electrons in GAMESS.

Continued code development is required in the module that computes the positron-electron annihilation observables. The primary focus should be the development of code to compute annihilation rates for multireference wavefunctions (*e.g.*, FCI and CASSCF). Also, the current code that calculates the annihilation rate does not use symmetry or orthogonality rules, and implementation of these techniques is necessary to treat larger systems. Finally, code for computing the TPMD should be implemented, since momentum distribution techniques in PAS are powerful tools for exploring positron chemistry and point defects in materials. The expressions for computing the TPMD are given in Sections A.8 and A.9 of Appendix A.

6.2.2 Future Experimental Efforts. Clearly, the PALS spectra in Figures 11 and 12 would benefit from further optimization. The primary focus should be on improving the anode signal from the PMTs. According to Becvar, [6] the manufacturing process for the XP2020Q PMTs (used in the current configuration) was changed in 1995, and the newer models suffer from a poor signal, which degrades their performance in PALS systems. Fortunately, three new BaF_2 detectors based on Hamamatsu H3378-50 PMTs have been received.

Since three detectors are now available, it is recommended that a "double-stop" setup is used, in which the 1.27 MeV "birth" gamma is detected by one PMT while a pair of 0.511 MeV annihilation photons are detected by the other two PMTs. In such a configuration, the two stop detectors are placed face-to-face on the opposite sides of the source-sample assembly. Also, Saito *et al.* [47] were able to achieve a time resolution of 119 ps with this setup by replacing the analog electronics with ultra-fast digitizers.

Finally, the PAS methods of measuring the TPMD (*e.g.*, ACAR and DBAR) are powerful tools for probing the electronic and positronic wavefunctions of annihilation sites. Comparison of a measured and calculated TPMD would be the ultimate test of the positronic NEO method.

6.2.3 Positronic Systems for Modeling and Experiment. As mentioned in Section 2.1.3, Barnes *et al.* have measured VFRs of CH_3X (X = H, F, Cl), benzene, and ammonia. These systems were chosen due to their highly symmetric nature. They are very attractive from a computational standpoint and are good candidates for the MRPT and CASSCF methods (see Section 6.2.1).

Once the recommendations presented in Section 6.2.1 are implemented, the groundwork will have been laid for investigating positrons localized in point defects in semiconductors and other condensed phase systems. Also, it will be possible to study positron binding to surfaces. These systems will require the use of the surface integrated molecular orbital/molecular mechanics (SIMOMM) method of Shoemaker *et al.* , [54] which is integrated into the GAMESS code. Although SIMOMM was developed specifically for the study of surface chemical systems, it is a general method, so application to vacancy defects in condensed matter is possible. SIMOMM is a model in which a subset of atoms within a large set of atoms are treated by QM (any method available in GAMESS), while the remaining atoms are treated by molecular mechanics (MM). In order to satisfy any dangling valences created at the boundary between the QM atoms and the total set of atoms, capping atoms (usually hydrogens) are connected by single bonds and included in the QM treatment. The positions of these capping atoms are optimized, allowing for more flexibility than in other hybrid methods (*e.g.*, Morokuma's integrated molecular orbital/molecular mechanics (IMOMM) approach). For modeling positrons localized in point defects or on surfaces, an appropriate NEO method would be used for the QM part of the calculation.

There are still several neutral atoms, X, that could bind a positron that have yet to be modeled. According to Schrader, [52], the most notable are X = Ti and X = Hf, which have

IPs very near the ionization energy of Ps.[1] As discussed in Section 2.2.4, there should be an accidental degeneracy between the e^+X and $Ps[X^+]$ wavefunctions for these systems, resulting in large positron binding energies for both. These systems can be modeled as five-particle system. Interestingly, one will need to account for relativistic effects for the core electrons. Since the positron is attracted to the relativistic electrons, perhaps the positron will need to be treated relativistically as well. Also, e^+Au did not show binding at the CI level. Perhaps relativistic effects must be included for this system in order to get a bound system.

[1] Ti has an IP of 6.825 eV, and Hf has an IP of 6.803 eV.

Appendix A. Electron-Positron Annihilation Observables from NEO-HF and NEO-MP2 Calculations

In this appendix, the expressions necessary for computing the electron-positron annihilation rate and TPMDs from NEO-HF and NEO-MP2 wavefunctions are developed. First, the NEO-HF wavefunction and NEO-MP2 wavefunctions and energy expressions are described. Then, some notational and methodological groundwork is presented by developing expressions for the NEO-HF electron-positron attraction energy and NEO-MP2 energy expressions. Finally, the derivations for the electron-positron annihilation rate and TPMD at the HF and MP2 level are given.

A.1 NEO-HF Wavefunctions

For nuclear-electronic orbital (positronic) Hartree-Fock [NEO-HF] theory, the total electron-positron wavefunction is represented by a product of single-configurational electronic and positronic wavefunctions:

$$\Psi(\mathbf{x}^{\mathrm{e}},\mathbf{x}^{\mathrm{p}}) = \Phi_0^{\mathrm{e}}(\mathbf{x}^{\mathrm{e}})\Phi_0^{\mathrm{p}}(\mathbf{x}^{\mathrm{p}}), \tag{70}$$

where $\Phi_0^{\mathrm{e}}(\mathbf{x}^{\mathrm{e}})$ and $\Phi_0^{\mathrm{p}}(\mathbf{x}^{\mathrm{p}})$ are antisymmetrized wave functions (determinants of orthonormal spin orbitals) representing the electrons and positrons, respectively. The vectors $\mathbf{x}^{\mathrm{e}}$ and $\mathbf{x}^{\mathrm{p}}$ are the position and spin coordinates of all of the electrons and positrons in the system, respectively.

So for an N_{e}-electron, N_{p}-positron system,

$$\Phi_0^{\mathrm{e}}(\mathbf{x}^{\mathrm{e}}) = \frac{1}{\sqrt{N_{\mathrm{e}}!}} \begin{vmatrix} \chi_1^{\mathrm{e}}(\mathbf{x}_1^{\mathrm{e}}) & \chi_2^{\mathrm{e}}(\mathbf{x}_1^{\mathrm{e}}) & \cdots & \chi_{N_{\mathrm{e}}}^{\mathrm{e}}(\mathbf{x}_1^{\mathrm{e}}) \\ \chi_1^{\mathrm{e}}(\mathbf{x}_2^{\mathrm{e}}) & \chi_2^{\mathrm{e}}(\mathbf{x}_2^{\mathrm{e}}) & \cdots & \chi_{N_{\mathrm{e}}}^{\mathrm{e}}(\mathbf{x}_2^{\mathrm{e}}) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1^{\mathrm{e}}(\mathbf{x}_{N_{\mathrm{e}}}^{\mathrm{e}}) & \chi_2^{\mathrm{e}}(\mathbf{x}_{N_{\mathrm{e}}}^{\mathrm{e}}) & \cdots & \chi_{N_{\mathrm{e}}}^{\mathrm{e}}(\mathbf{x}_{N_{\mathrm{e}}}^{\mathrm{e}}) \end{vmatrix}, \tag{71}$$

and

$$\Phi_0^{\mathrm{p}}(\mathbf{x}^{\mathrm{p}}) = \frac{1}{\sqrt{N_{\mathrm{p}}!}} \begin{vmatrix} \chi_1^{\mathrm{p}}(\mathbf{x}_1^{\mathrm{p}}) & \chi_2^{\mathrm{p}}(\mathbf{x}_1^{\mathrm{p}}) & \cdots & \chi_{N_{\mathrm{p}}}^{\mathrm{p}}(\mathbf{x}_1^{\mathrm{p}}) \\ \chi_1^{\mathrm{p}}(\mathbf{x}_2^{\mathrm{p}}) & \chi_2^{\mathrm{p}}(\mathbf{x}_2^{\mathrm{p}}) & \cdots & \chi_{N_{\mathrm{p}}}^{\mathrm{p}}(\mathbf{x}_2^{\mathrm{p}}) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1^{\mathrm{p}}(\mathbf{x}_{N_{\mathrm{p}}}^{\mathrm{p}}) & \chi_2^{\mathrm{p}}(\mathbf{x}_{N_{\mathrm{p}}}^{\mathrm{p}}) & \cdots & \chi_{N_{\mathrm{p}}}^{\mathrm{p}}(\mathbf{x}_{N_{\mathrm{p}}}^{\mathrm{p}}) \end{vmatrix}, \tag{72}$$

where

$$\mathbf{x}^{\mathrm{e}} = \{\mathbf{x}_1^{\mathrm{e}}, \mathbf{x}_2^{\mathrm{e}}, ..., \mathbf{x}_{N_{\mathrm{e}}}^{\mathrm{e}}\} \tag{73}$$

$$\mathbf{x}^{\mathrm{p}} = \{\mathbf{x}_1^{\mathrm{p}}, \mathbf{x}_2^{\mathrm{p}}, ..., \mathbf{x}_{N_{\mathrm{p}}}^{\mathrm{p}}\} \tag{74}$$

Note that each χ is a product of a spatial function (ϕ) and a spin (α or β) function: For example,

$$\chi_1^{\mathrm{e}}(\mathbf{x}_1^{\mathrm{e}}) = \phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\alpha(\omega_1^{\mathrm{e}}) \tag{75}$$

$$\chi_2^{\mathrm{e}}(\mathbf{x}_2^{\mathrm{e}}) = \phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\beta(\omega_2^{\mathrm{e}}) \tag{76}$$

In all of our calculations, the electronic part is treated at the RHF level. This allows the electronic wave function to be written as

$$\Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}}) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}}) & \phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}}) & \cdots & \phi_N^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}}) \\ \phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}}) & \phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}}) & \cdots & \phi_N^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1^{\mathrm{e}}(\mathbf{r}_N^{\mathrm{e}}) & \phi_2^{\mathrm{e}}(\mathbf{r}_N^{\mathrm{e}}) & \cdots & \phi_N^{\mathrm{e}}(\mathbf{r}_N^{\mathrm{e}}) \end{vmatrix}, \tag{77}$$

where N is the number of doubly-occupied spatial orbitals. Also, in all of the systems of interest, there is only one positron present. Therefore, the HF ground state positronic determinant is simply

$$\Phi_0^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) = \phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}), \tag{78}$$

with the subscript on the positronic coordinate absent, and the NEO-HF wavefunction is reduced to

$$\Psi^{\mathrm{HF}} = \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \tag{79}$$

Each spatial orbital is a linear combination of Gaussian primitives,

$$\phi_i = \sum_i C_i \varphi_i^{l_i,m_i,n_i}(\alpha_i, \mathbf{A}_i), \tag{80}$$

where $\varphi_i^{l_i,m_i,n_i}(\alpha_i, \mathbf{A}_i)$ is defined by

$$\begin{aligned}
\varphi_i^{l_i,m_i,n_i}(\alpha_i, \mathbf{A}_i) &= x_{\mathbf{A}_i}^{l_i} y_{\mathbf{A}_i}^{m_i} z_{\mathbf{A}_i}^{n_i} \varphi_i^0(\alpha_i, \mathbf{A}_i), && (81)\\
\varphi_i^0(\alpha_i, \mathbf{A}_i) &= \exp\left[-\alpha_i(x_{\mathbf{A}_i}^2 + y_{\mathbf{A}_i}^2 + z_{\mathbf{A}_i}^2)\right], && (82)\\
x_{\mathbf{A}_i} &= (x - \mathbf{A}_{i,x}), && (83)\\
y_{\mathbf{A}_i} &= (y - \mathbf{A}_{i,y}), && (84)\\
z_{\mathbf{A}_i} &= (z - \mathbf{A}_{i,z}), && (85)
\end{aligned}$$

and, $\mathbf{A}_i = (\mathbf{A}_{i,x}, \mathbf{A}_{i,y}, \mathbf{A}_{i,z})$ are the coordinates for the point about which the function is located. The polynomial exponents, $\{l_i, m_i, n_i\} \in \{0, 1, 2, \ldots\}$, can be summed to yield the angular momentum of the Gaussian function ($\lambda_i = l_i + m_i + n_i$). The Gaussian exponent, α_i, is a positive number.

A.2 NEO-MP2 Wavefunctions and Energies

In this section, the method for calculating the NEO-MP2 energy and wavefunction is shown. The general derivation of the NEO many-body perturbation theory is given in [60]. For a system with N_e paired or high-spin electrons and N_p paired or high-spin positrons, the NEO-HF Hamiltonian is

$$\begin{aligned} H_0^{HF} = &\sum_i^{N_e} \left\{ h^e(i) + \nu_{ee}^{HF}(i) + \nu_{ep,e}^{HF}(i) \right\} \\ &+ \sum_{i'}^{N_p} \left\{ h^p(i') + \nu_{pp}^{HF}(i') + \nu_{ep,p}^{HF}(i') \right\} \end{aligned} \tag{86}$$

The unprimed indices refer to electrons, and the primed indices refer to positrons. The superscripts/subscripts 'e' and 'p' on the operators denote electrons and positrons, respectively. In this expression, h^e and h^p are the one particle terms and are defined in ref. [71].

The two-particle operators are defined as:

$$\begin{aligned} \nu_{ee}^{HF}(1)\chi_j^e(1) = &\sum_b \left\langle \chi_b^e(2) \middle| r_{12}^{-1} \middle| \chi_b^e(2) \right\rangle \chi_j^e(1) \\ &- \sum_b \left\langle \chi_b^e(2) \middle| r_{12}^{-1} \middle| \chi_j^e(2) \right\rangle \chi_b^e(1) \end{aligned} \tag{87}$$

$$\begin{aligned} \nu_{pp}^{HF}(1')\chi_{j'}^p(1') = &\sum_{b'} \left\langle \chi_{b'}^e(2') \middle| r_{1'2'}^{-1} \middle| \chi_{b'}^e(2') \right\rangle \chi_{j'}^e(1') \\ &- \sum_{b'} \left\langle \chi_{b'}^e(2') \middle| r_{1'2'}^{-1} \middle| \chi_{j'}^e(2') \right\rangle \chi_{b'}^e(1') \end{aligned} \tag{88}$$

$$\nu_{ep,e}^{HF}(1)\chi_j^e(1) = -\sum_{b'} \left\langle \chi_{b'}^p(1') \middle| r_{11'}^{-1} \middle| \chi_{b'}^p(1') \right\rangle \chi_j^e(1) \tag{89}$$

$$\nu_{ep,p}^{HF}(1')\chi_{j'}^p(1') = -\sum_b \left\langle \chi_b^e(1') \middle| r_{11'}^{-1} \middle| \chi_b^e(1) \right\rangle \chi_{j'}^p(1') \tag{90}$$

where the spin orbitals are referred to by the symbol χ, and the summations $\sum_b$ and $\sum_{b'}$ are over the occupied electronic and positronic spin orbitals, respectively. Thus, $\nu_{ee}^{HF}(i)$ and $\nu_{pp}^{HF}(i')$ are the Coulomb-exchange operators for the electrons and positrons, respectively, and $\nu_{ep,e}^{HF}(i)$ and $\nu_{ep,p}^{HF}(i')$ are the electron-positron Coulomb operators for the electrons and

positrons, respectively. It can then be shown that the orbital energies can be expressed as

$$\varepsilon_i^{\mathrm{e}} = \langle i|h^{\mathrm{e}}|i\rangle + \sum_b \left[\langle ib\,|\,ib\rangle - \langle ib\,|\,bi\rangle\right] - \sum_{b'} \left\langle ib' \,\middle|\, ib' \right\rangle, \tag{91}$$

$$\varepsilon_{i'}^{\mathrm{p}} = \left\langle i' \middle| h^{\mathrm{p}} \middle| i' \right\rangle + \sum_{b'} \left[\left\langle i'b' \,\middle|\, i'b' \right\rangle - \left\langle i'b' \,\middle|\, b'i' \right\rangle\right] - \sum_{b'} \left\langle bi' \,\middle|\, bi' \right\rangle, \tag{92}$$

where

$$\langle ij\,|\,kl\rangle = \int d\mathbf{x}_1 d\mathbf{x}_2 \chi_i^{\mathrm{e}*}(\mathbf{x}_1)\chi_j^{\mathrm{e}*}(\mathbf{x}_2)\frac{1}{r_{12}}\chi_k^{\mathrm{e}}(\mathbf{x}_1)\chi_l^{\mathrm{e}}(\mathbf{x}_2), \tag{93}$$

$$\left\langle ij' \,\middle|\, kl' \right\rangle = \int d\mathbf{x}_1 d\mathbf{x}_{1'} \chi_i^{\mathrm{e}*}(\mathbf{x}_1)\chi_{j'}^{\mathrm{p}*}(\mathbf{x}_{1'})\frac{1}{r_{12}}\chi_k^{\mathrm{e}}(\mathbf{x}_1)\chi_{l'}^{\mathrm{p}}(\mathbf{x}_{1'}) \tag{94}$$

The perturbations for electron-electron, positron-positron, and electron-positron correlation are

$$W_{\mathrm{ee}} = \sum_{i<j} \frac{1}{r_{ij}} - \sum_i \nu_{\mathrm{ee}}^{\mathrm{HF}}(i), \tag{95}$$

$$W_{\mathrm{pp}} = \sum_{i'<j'} \frac{1}{r_{i'j'}} - \sum_{i'} \nu_{\mathrm{pp}}^{\mathrm{HF}}(i'), \tag{96}$$

$$W_{\mathrm{ep}} = -\sum_{i,i'} \frac{1}{r_{ii'}} - \sum_i \nu_{\mathrm{ep,e}}^{\mathrm{HF}}(i) - \sum_{i'} \nu_{\mathrm{ep,p}}^{\mathrm{HF}}(i') \tag{97}$$

Here, and in subsequent equations, the upper bounds for summations over unprimed and primed particle indices are N_{e} and N_{p}, respectively. The combined perturbation for electron-electron, positron-positron, and electron-positron correlation is $W = W_{\mathrm{ee}} + W_{\mathrm{pp}} + W_{\mathrm{ep}}$.

Using Rayleigh-Schrödinger perturbation theory, the NEO-MP2 (second-order many-body perturbation theory) energy expression for mixed electronic-positronic wavefunctions is

$$E_{00}^{(2)} = \sum_{(nn')\neq(00)} \frac{\left|\left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| W \middle| \Phi_n^{\mathrm{e}}\Phi_{n'}^{\mathrm{p}} \right\rangle\right|^2}{E_{00}^{(0)} - E_{nn'}^{(0)}} \tag{98}$$

Here, the zeroth-order wavefunctions (*i.e.*, the eigenfunctions of H_0^{HF} given in equation 86) are $\left|\Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}}\right\rangle$ for the ground state and $\left|\Phi_n^{\mathrm{e}}\Phi_{n'}^{\mathrm{p}}\right\rangle$ for excited states, using Dirac notation for the

Slater determinant of spin orbitals:

$$|\Phi_0^{\mathrm{e}}\rangle = \left|\chi_1^{\mathrm{e}}(1)\chi_2^{\mathrm{e}}(2)\cdots\chi_{N_{\mathrm{e}}}^{\mathrm{e}}(N_{\mathrm{e}})\right\rangle \tag{99}$$

The Dirac forms of the singly- and doubly-substituted Slater determinants of spin orbitals are

$$|\Phi_a^r\rangle = \left|\chi_1^{\mathrm{e}}(1)\chi_2^{\mathrm{e}}(2)\cdots\chi_{a-1}^{\mathrm{e}}(a-1)\chi_r^{\mathrm{e}}(a)\chi_{a+1}^{\mathrm{e}}(a+1)\cdots\cdots\chi_{N_{\mathrm{e}}}^{\mathrm{e}}(N_{\mathrm{e}})\right\rangle, \tag{100}$$

$$\begin{aligned}|\Phi_{ab}^{rs}\rangle = \Big|&\chi_1^{\mathrm{e}}(1)\chi_2^{\mathrm{e}}(2)\cdots \\ &\cdots\chi_{a-1}^{\mathrm{e}}(a-1)\chi_r^{\mathrm{e}}(a)\chi_{a+1}^{\mathrm{e}}(a+1)\cdots \\ &\cdots\chi_{b-1}^{\mathrm{e}}(b-1)\chi_s^{\mathrm{e}}(b)\chi_{b+1}^{\mathrm{e}}(b+1)\cdots\chi_{N_{\mathrm{e}}}^{\mathrm{e}}(N_{\mathrm{e}})\Big\rangle\end{aligned} \tag{101}$$

The zeroth-order energy $E_{nn'}^{(0)}$ is the sum of the energies corresponding to the occupied electronic and positronic spin orbitals for the state $|\Phi_n^{\mathrm{e}}\Phi_{n'}^{\mathrm{p}}\rangle$. The notation under the summation in Equation (98) indicates that both n and n' cannot simultaneously correspond to the ground state electronic and positronic determinants, respectively. The first-order correction to the HF wavefunction is

$$\Psi_{00}^{(1)} = \sum_{(nn')\neq(00)} \frac{\left\langle\Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}}\middle|W\middle|\Phi_n^{\mathrm{e}}\Phi_{n'}^{\mathrm{p}}\right\rangle}{E_{00}^{(0)} - E_{nn'}^{(0)}} |\Phi_n^{\mathrm{e}}\Phi_{n'}^{\mathrm{p}}\rangle \tag{102}$$

Substituting $W = W_{\mathrm{ee}} + W_{\mathrm{pp}} + W_{\mathrm{ep}}$ into Equations (98) and (102), the second-order energy correction and the first-order wavefunction correction within the NEO framework can be expressed as

$$\begin{aligned}E_{00}^{(2)} = &\sum_{n\neq 0} \frac{\left|\left\langle\Phi_0^{\mathrm{e}}\middle|W_{\mathrm{ee}}\middle|\Phi_n^{\mathrm{e}}\right\rangle\right|^2}{E_0^{\mathrm{e}} - E_n^{\mathrm{e}}} + \sum_{n'\neq 0} \frac{\left|\left\langle\Phi_0^{\mathrm{p}}\middle|W_{\mathrm{pp}}\middle|\Phi_{n'}^{\mathrm{p}}\right\rangle\right|^2}{E_0^{\mathrm{p}} - E_{n'}^{\mathrm{p}}} \\ &+ \sum_{(nn')\neq(00)} \frac{\left|\left\langle\Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}}\middle|W_{\mathrm{ep}}\middle|\Phi_n^{\mathrm{e}}\Phi_{n'}^{\mathrm{p}}\right\rangle\right|^2}{E_0^{\mathrm{e}} + E_0^{\mathrm{p}} - E_n^{\mathrm{e}} - E_{n'}^{\mathrm{p}}}\end{aligned} \tag{103}$$

and

$$\Psi_{00}^{(1)} = \sum_{n\neq 0} \frac{\left\langle \Phi_0^{\mathrm{e}} \middle| W_{\mathrm{ee}} \middle| \Phi_n^{\mathrm{e}} \right\rangle}{E_0^{\mathrm{e}} - E_n^{\mathrm{e}}} \left| \Phi_n^{\mathrm{e}} \Phi_0^{\mathrm{p}} \right\rangle + \sum_{n'\neq 0} \frac{\left\langle \Phi_0^{\mathrm{p}} \middle| W_{\mathrm{pp}} \middle| \Phi_{n'}^{\mathrm{p}} \right\rangle}{E_0^{\mathrm{p}} - E_{n'}^{\mathrm{p}}} \left| \Phi_0^{\mathrm{e}} \Phi_{n'}^{\mathrm{p}} \right\rangle + \sum_{(nn')\neq(00)} \frac{\left\langle \Phi_0^{\mathrm{e}} \Phi_0^{\mathrm{p}} \middle| W_{\mathrm{ep}} \middle| \Phi_n^{\mathrm{e}} \Phi_{n'}^{\mathrm{p}} \right\rangle}{E_0^{\mathrm{e}} + E_0^{\mathrm{p}} - E_n^{\mathrm{e}} - E_{n'}^{\mathrm{p}}} \left| \Phi_n^{\mathrm{e}} \Phi_{n'}^{\mathrm{p}} \right\rangle, \tag{104}$$

respectively. The W_{ee} terms become the standard MP2 equations for electron-electron correlation in electronic structure theory

$$E_{\mathrm{ee}}^{(2)} = \sum_{\substack{a<b \\ r<s}} \frac{\left| \langle ab|rs\rangle - \langle ab|sr\rangle \right|^2}{\varepsilon_a^{\mathrm{e}} + \varepsilon_b^{\mathrm{e}} - \varepsilon_r^{\mathrm{e}} - \varepsilon_s^{\mathrm{e}}} \tag{105}$$

and

$$\Psi_{\mathrm{ee}}^{(1)} = \sum_{\substack{a<b \\ r<s}} \frac{\langle ab|rs\rangle - \langle ab|sr\rangle}{\varepsilon_a^{\mathrm{e}} + \varepsilon_b^{\mathrm{e}} - \varepsilon_r^{\mathrm{e}} - \varepsilon_s^{\mathrm{e}}} \left| \Phi_{ab}^{rs} \Phi_0^{\mathrm{p}} \right\rangle \tag{106}$$

$$= \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C_{ab}^{rs} \left| \Phi_{ab}^{rs} \Phi_0^{\mathrm{p}} \right\rangle, \tag{107}$$

where a and b refer to occupied electronic spin orbitals, and r and s refer to virtual electronic spin orbitals.

The W_{pp} terms are

$$E_{\mathrm{pp}}^{(2)} = \sum_{\substack{a'<b' \\ r'<s'}} \frac{\left| \langle a'b'|r's'\rangle - \langle a'b'|s'r'\rangle \right|^2}{\varepsilon_{a'}^{\mathrm{p}} + \varepsilon_{b'}^{\mathrm{p}} - \varepsilon_{r'}^{\mathrm{p}} - \varepsilon_{s'}^{\mathrm{p}}} \tag{108}$$

and

$$\Psi_{\mathrm{pp}}^{(1)} = \sum_{\substack{a'<b' \\ r'<s'}} \frac{\langle a'b'|r's'\rangle - \langle a'b'|s'r'\rangle}{\varepsilon_{a'}^{\mathrm{p}} + \varepsilon_{b'}^{\mathrm{p}} - \varepsilon_{r'}^{\mathrm{p}} - \varepsilon_{s'}^{\mathrm{p}}} \left| \Phi_0^{\mathrm{e}} \Phi_{a'b'}^{r's'} \right\rangle \tag{109}$$

$$= \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{pp}}C_{a'b'}^{r's'} \left| \Phi_0^{\mathrm{e}} \Phi_{a'b'}^{r's'} \right\rangle, \tag{110}$$

where a' and b' refer to occupied positronic spin orbitals, and r' and s' refer to virtual positronic spin orbitals. The only non-zero matrix elements for $\left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}}\middle|W_{\mathrm{ep}}\middle|\Phi_n^{\mathrm{e}}\Phi_{n'}^{\mathrm{p}}\right\rangle$ include single excitations in both the electron and positron determinants, leading to the second-order electron-positron correlation correction to the energy (see Section A.4 for the derivation)

$$E_{\mathrm{ep}}^{(2)} = \sum_{aa'rr'} \frac{|\langle aa'|rr'\rangle|^2}{\varepsilon_a^{\mathrm{e}} + \varepsilon_{a'}^{\mathrm{p}} - \varepsilon_r^{\mathrm{e}} - \varepsilon_{r'}^{\mathrm{p}}} \tag{111}$$

and the first-order electron-positron correlation correction to the wavefunction

$$\Psi_{\mathrm{ep}}^{(1)} = -\sum_{aa'rr'} \frac{\langle aa'|rr'\rangle}{\varepsilon_a^{\mathrm{e}} + \varepsilon_{a'}^{\mathrm{p}} - \varepsilon_r^{\mathrm{e}} - \varepsilon_{r'}^{\mathrm{p}}} |\Phi_a^r \Phi_{a'}^{r'}\rangle \tag{112}$$

$$= -\sum_{aa'rr'} {}^{\mathrm{ep}}C_{aa'}^{rr'} |\Phi_a^r \Phi_{a'}^{r'}\rangle, \tag{113}$$

where a and a' denote occupied electronic and positronic spin orbitals, respectively, and r and r' denote virtual electronic and positronic spin orbitals, respectively. The total NEO-MP2 energy is $E^{\mathrm{MP2}} = E^{\mathrm{HF}} + E_{\mathrm{ee}}^{(2)} + E_{\mathrm{pp}}^{(2)} + E_{\mathrm{ep}}^{(2)}$.

For systems with only one positron, $E_{\mathrm{pp}}^{(2)}$ and $\Psi_{\mathrm{pp}}^{(1)}$ vanish, and the electron-positron corrections can be simplified to

$$E_{\mathrm{ep}}^{(2)} = \sum_{arr'} \frac{|\langle a1'|rr'\rangle|^2}{\varepsilon_a^{\mathrm{e}} + \varepsilon_{1'}^{\mathrm{p}} - \varepsilon_r^{\mathrm{e}} - \varepsilon_{r'}^{\mathrm{p}}} \tag{114}$$

and

$$\Psi_{\mathrm{ep}}^{(1)} = \sum_{arr'} \frac{\langle a1'|rr'\rangle}{\varepsilon_a^{\mathrm{e}} + \varepsilon_{1'}^{\mathrm{p}} - \varepsilon_r^{\mathrm{e}} - \varepsilon_{r'}^{\mathrm{p}}} |\Phi_a^r \Phi_{1'}^{r'}\rangle \tag{115}$$

$$= \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} |\Phi_a^r \Phi_{1'}^{r'}\rangle \tag{116}$$

The resulting NEO-MP2 energy is

$$E^{\mathrm{MP2}} = E^{\mathrm{HF}} + \sum_{\substack{a<b \\ r<s}} \frac{|\langle ab|rs\rangle - \langle ab|sr\rangle|^2}{\varepsilon_a^{\mathrm{e}} + \varepsilon_b^{\mathrm{e}} - \varepsilon_r^{\mathrm{e}} - \varepsilon_s^{\mathrm{e}}} + \sum_{arr'} \frac{|\langle a1'|rr'\rangle|^2}{\varepsilon_a^{\mathrm{e}} + \varepsilon_{1'}^{\mathrm{p}} - \varepsilon_r^{\mathrm{e}} - \varepsilon_{r'}^{\mathrm{p}}}, \tag{117}$$

and the first-order wavefunction correction is

$$\Psi_{00}^{(1)} = \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C_{ab}^{rs} \, |\Phi_{ab}^{rs}\Phi_{0}^{\mathrm{p}}\rangle - \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \, |\Phi_{a}^{r}\Phi_{1'}^{r'}\rangle \,, \tag{118}$$

where

$${}^{\mathrm{ee}}C_{ab}^{rs} = \frac{\langle ab|rs\rangle - \langle ab|sr\rangle}{\varepsilon_a^{\mathrm{e}} + \varepsilon_b^{\mathrm{e}} - \varepsilon_r^{\mathrm{e}} - \varepsilon_s^{\mathrm{e}}}, \tag{119}$$

$${}^{\mathrm{ep}}C_{a1'}^{rr'} = \frac{\langle a1'|rr'\rangle}{\varepsilon_a^{\mathrm{e}} + \varepsilon_{1'}^{\mathrm{p}} - \varepsilon_r^{\mathrm{e}} - \varepsilon_{r'}^{\mathrm{p}}} \tag{120}$$

A.3 NEO-HF Electron-Positron Attraction Energy

The electron-positron attraction energy for the NEO-HF wavefunction, given in equation (79), consisting of N doubly-occupied electronic orbitals and a single positron is

$$\left\langle \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \right| -2\sum_{k}^{N} r_{k1'}^{-1} \left| \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \right\rangle \tag{121}$$

All of our wavefunctions consist of real spatial functions, and the complex conjugate notation can be dropped. For example:

$$\Phi_0^{\mathrm{e}*} = \Phi_0^{\mathrm{e}} \tag{122}$$

$$\langle \Phi_0^{\mathrm{e}} | = | \Phi_0^{\mathrm{e}} \rangle \tag{123}$$

By simplifying the expression in equation (121), the utility of expanding a determinant in minors will be presented. Further information on this concept may be found in Appendix 2.A.3 of Cook. [10]

We can expand a determinant of order n in terms of the first column to show its dependence on the coordinates of particle 1 explicitly:

$$\Phi(\mathbf{r}_1,\mathbf{r}_2,...\mathbf{r}_n)=\frac{1}{\sqrt{n!}}\begin{vmatrix}\phi_1(\mathbf{r}_1) & \phi_1(\mathbf{r}_2) & \cdots & \phi_1(\mathbf{r}_n)\\ \phi_2(\mathbf{r}_1) & \phi_2(\mathbf{r}_2) & \cdots & \phi_2(\mathbf{r}_n)\\ \phi_3(\mathbf{r}_1) & \phi_3(\mathbf{r}_2) & \cdots & \phi_3(\mathbf{r}_n)\\ \vdots & \vdots & \ddots & \vdots\\ \phi_{n-2}(\mathbf{r}_1) & \phi_{n-2}(\mathbf{r}_2) & \cdots & \phi_{n-2}(\mathbf{r}_n)\\ \phi_{n-1}(\mathbf{r}_1) & \phi_{n-1}(\mathbf{r}_2) & \cdots & \phi_{n-1}(\mathbf{r}_n)\\ \phi_n(\mathbf{r}_1) & \phi_n(\mathbf{r}_2) & \cdots & \phi_n(\mathbf{r}_n)\end{vmatrix}$$

$$=\frac{1}{\sqrt{n!}}\sum_{i=1}^{n}(-1)^{i+1}\phi_i(\mathbf{r}_1)\begin{vmatrix}\phi_1(\mathbf{r}_2) & \phi_1(\mathbf{r}_3) & \cdots & \phi_1(\mathbf{r}_n)\\ \phi_2(\mathbf{r}_2) & \phi_2(\mathbf{r}_3) & \cdots & \phi_2(\mathbf{r}_n)\\ \phi_3(\mathbf{r}_2) & \phi_3(\mathbf{r}_3) & \cdots & \phi_3(\mathbf{r}_n)\\ \vdots & \vdots & \ddots & \vdots\\ \phi_{i-1}(\mathbf{r}_2) & \phi_{i-1}(\mathbf{r}_3) & \cdots & \phi_{i-1}(\mathbf{r}_n)\\ \phi_{i+1}(\mathbf{r}_2) & \phi_{i+1}(\mathbf{r}_3) & \cdots & \phi_{i+1}(\mathbf{r}_n)\\ \vdots & \vdots & \ddots & \vdots\\ \phi_{n-2}(\mathbf{r}_2) & \phi_{n-2}(\mathbf{r}_3) & \cdots & \phi_{n-2}(\mathbf{r}_n)\\ \phi_{n-1}(\mathbf{r}_2) & \phi_{n-1}(\mathbf{r}_3) & \cdots & \phi_{n-1}(\mathbf{r}_n)\\ \phi_n(\mathbf{r}_2) & \phi_n(\mathbf{r}_3) & \cdots & \phi_n(\mathbf{r}_n)\end{vmatrix}$$

$$=\frac{1}{\sqrt{n!}}\sum_{i=1}^{n}(-1)^{i+1}\phi_i(\mathbf{r}_1)\Phi_{1i} \tag{124}$$

where Φ_{1i} is the determinant of order n-1 obtained by striking out the i-th row and first column of Φ but without the factor of $1/\sqrt{(n-1)!}$. This means that Φ_{1i} is not normalized.

We also define $d\mathbf{r}^{\mathrm{e}}_{(k)}$ as

$$d\mathbf{r}^{\mathrm{e}}_{(k)}=d\mathbf{r}^{\mathrm{e}}_1 d\mathbf{r}^{\mathrm{e}}_2\cdots d\mathbf{r}^{\mathrm{e}}_{(k-1)}d\mathbf{r}^{\mathrm{e}}_{(k+1)}\cdots d\mathbf{r}^{\mathrm{e}}_N \tag{125}$$

The interdeterminant overlap integral is defined as the determinant of the overlap matrix between the ki and kj basis.

$$\int \Phi^{\mathrm{e}}_{0\,ki}\,\Phi^{\mathrm{e}}_{0\,kj}\,d\mathbf{r}^{\mathrm{e}}_{(k)} = \frac{det|S(ki;kj)|}{\sqrt{D(ki)D(kj)}} \tag{126}$$

$D(ki)$ and $D(kj)$ are just the determinants of the overlap matrix for the ki and kj basis, respectively:

$$D(ki) = D(kj) = 1 \tag{127}$$

since we have an orthonormal set of orbitals, and

$$\int \Phi^{\mathrm{e}}_{0\,ki}\,\Phi^{\mathrm{e}}_{0\,kj}\,d\mathbf{r}^{\mathrm{e}}_{(k)} = \frac{det|S(ki;kj)|}{\cancelto{1}{\sqrt{D^{ki}D^{kj}}}} \tag{128}$$

$$= det|S(ki;kj)| \tag{129}$$

$|S(ki;kj)|$ is simply the determinant of the MO overlap matrix with the i-th row and j-th column removed:

$$\begin{vmatrix} S_{11} & S_{12} & \cdots & S_{1(j-1)} & S_{1(j+1)} & \cdots & S_{1N} \\ S_{21} & S_{22} & \cdots & S_{2(j-1)} & S_{2(j+1)} & \cdots & S_{2N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ S_{(i-1)1} & S_{(i-1)2} & \cdots & S_{(i-1)(j-1)} & S_{(i-1)(j+1)} & \cdots & S_{(i-1)N} \\ S_{(i+1)1} & S_{(i+1)2} & \cdots & S_{(i+1)(j-1)} & S_{(i+1)(j+1)} & \cdots & S_{(i+1)N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ S_{N1} & S_{N2} & \cdots & S_{N(j-1)} & S_{N(j+1)} & \cdots & S_{NN} \end{vmatrix} \tag{130}$$

where

$$S_{ij} = \begin{cases} 0\,, i \neq j \\ 1\,, i = j \end{cases} \tag{131}$$

Thus, if $i = j$, the above determinant is one; however, if $i \neq j$, there is at least one zero on the diagonal, and the above determinant is zero. So, the interdeterminant overlap integral

is then

$$\int \Phi^{\mathrm{e}}_{0\,ki}\,\Phi^{\mathrm{e}}_{0\,kj}\,dr^{\mathrm{e}}_{(k)} = \begin{cases} 0 & i \neq j, \\ (N-1)! & i = j. \end{cases} \tag{132}$$

$$= (N-1)!\delta_{ij} \tag{133}$$

and

$$\sum_{k=1}^{N} \int \Phi^{\mathrm{e}}_{0\,ki}\,\Phi^{\mathrm{p}}_{0\,kj}\,d\mathbf{r}^{\mathrm{e}}_{(k)} = N!\delta_{ij}, \tag{134}$$

and

$$\int \Phi_{1i}\Phi_{1i} = (n-1)! \tag{135}$$

Now the electron-positron attraction energy expression can be simplified:

$$\left\langle \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \right| -2\sum_k^N r_{k1'}^{-1} \left| \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \right\rangle \tag{136}$$

$$= -2\sum_k^N \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) r_{k1'}^{-1} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \tag{137}$$

$$= -2N \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \tag{138}$$

$$\begin{aligned} &= -\frac{2N}{N!}\sum_{ij}(-1)^{i+j} \\ &\quad \times \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1) \Phi_{\substack{0\\1i}}^{\mathrm{e}} \phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \phi_j(\mathbf{r}_1) \Phi_{\substack{0\\1j}}^{\mathrm{e}} \phi_{1'}(\mathbf{r}_{1'}) \end{aligned} \tag{139}$$

$$\begin{aligned} &= -\frac{2}{(N-1)!}\sum_{ij}(-1)^{i+j} \int d\mathbf{r}_{(1)}^{\mathrm{e}} \Phi_{\substack{0\\1i}}^{\mathrm{e}} \Phi_{\substack{0\\1j}}^{\mathrm{e}} \\ &\quad \times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \phi_j(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) \end{aligned} \tag{140}$$

$$\begin{aligned} &= -\frac{2}{(N-1)!}\sum_{ij}(-1)^{i+j}\delta_{ij} \\ &\quad \times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \phi_j(\mathbf{r}_1)\phi_{j'}(\mathbf{r}_{1'}) \end{aligned} \tag{141}$$

$$\begin{aligned} &= -\frac{2(N-1)!}{(N-1)!}\sum_{i}(-1)^{2i} \\ &\quad \times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{i'}(\mathbf{r}_{1'}) r_{11'}^{-1} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1) \end{aligned} \tag{142}$$

$$= -2\sum_i \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{i'}(\mathbf{r}_{1'}) r_{11'}^{-1} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) \tag{143}$$

$$= -2\sum_i \left\langle i1' \middle| i1' \right\rangle \tag{144}$$

If we let β (β') be the number of electronic (positronic) basis functions, this expression can be expanded in terms of the electron and positron basis functions:

$$\begin{aligned} -2\sum_{i}^{N}\left\langle i1' \,\middle|\, i1' \right\rangle &= -2\sum_{i}^{N}\int d\mathbf{r}_1 d\mathbf{r}_{1'} \sum_{m}^{\beta} c_{im}\varphi_m(\mathbf{r}_1)\sum_{m'}^{\beta'} c_{1'm'}\varphi_{m'}(\mathbf{r}_{1'}) \\ &\quad \times r_{11'}^{-1}\sum_{n}^{\beta} c_{in}\varphi_n(\mathbf{r}_1)\sum_{n'}^{\beta'} c_{1'n'}\varphi_{n'}(\mathbf{r}_{1'}) \end{aligned} \tag{145}$$

$$\begin{aligned} &= -2\sum_{i}^{N}\sum_{mn}^{\beta}\sum_{m'n'}^{\beta'} c_{im}c_{in}c_{1'm'}c_{1'n'} \\ &\quad \times \int d\mathbf{r}_1 d\mathbf{r}_{1'}\varphi_m(\mathbf{r}_1)\varphi_{m'}(\mathbf{r}_{1'})r_{11'}^{-1}\varphi_n(\mathbf{r}_1)\varphi_{n'}(\mathbf{r}_{1'}) \end{aligned} \tag{146}$$

$$\begin{aligned} &= -2\sum_{i}^{N}\sum_{m}^{\beta}\sum_{n}^{m}\zeta_{mn}\sum_{m'}^{\beta'}\sum_{n'}^{m'}\zeta_{m'n'}c_{im}c_{in}c_{1'm'}c_{1'n'} \\ &\quad \times \int d\mathbf{r}_1 d\mathbf{r}_{1'}\varphi_m(\mathbf{r}_1)\varphi_{m'}(\mathbf{r}_{1'})r_{11'}^{-1}\varphi_n(\mathbf{r}_1)\varphi_{n'}(\mathbf{r}_{1'}) \end{aligned} \tag{147}$$

$$= -2\sum_{i}^{N}\sum_{m}^{\beta}\sum_{n}^{m}\sum_{m'}^{\beta'}\sum_{n'}^{m'}\zeta_{mn}\zeta_{m'n'}c_{im}c_{in}c_{1'm'}c_{1'n'}\left\langle mm' \,\middle|\, nn' \right\rangle \tag{148}$$

where

$$\zeta_{mn} = \begin{cases} 1 & m = n \\ 2 & m \neq n \end{cases} \tag{149}$$

and c_{im} is the ith MO coefficient for the mth atomic orbital (AO) basis function. $\langle mm' \,|\, nn' \rangle$ is an electron-positron attraction integral in the AO basis.

A.4 W_{ep} Matrix Elements

Now, we turn our attention to the NEO-MP2 electron-positron correlation correction to show how one arrives at equation (111) We will examine the single-electron excitation, the single-positron excitation, and the single-electron/single-positron excitation individually.

A.4.1 Single-Electron Excitation. First, we compute the matrix element of the W_{ep} operator, equation (97), between the NEO-HF ground state wavefunction and the wavefunction consisting of a single-electronic excitation:

$$\left\langle \Phi_0^{\text{e}}\Phi_0^{\text{p}}\middle|\hat{W}_{\text{ep}}\middle|\Phi_a^r\Phi_0^{\text{p}}\right\rangle = \left\langle \Phi_0^{\text{e}}\Phi_0^{\text{p}}\middle| \overbrace{-\sum_{kk'} r_{kk'}^{-1}}^{\text{Term 1}} \overbrace{-\sum_{k} \nu_{\text{ep,e}}^{\text{HF}}(k)}^{\text{Term 2}} \overbrace{-\sum_{k'} \nu_{\text{ep,p}}^{\text{HF}}(k')}^{\text{Term 3}} \middle|\Phi_a^r\Phi_0^{\text{p}}\right\rangle \tag{150}$$

A definition to help with the derivation:

$$\underset{i}{\Psi}{}_{abcd\cdots}^{rstu\cdots} = \begin{cases} \phi_i & i \neq a, i \neq b, i \neq c, i \neq d, \ldots \\ \phi_r & i = a, \\ \phi_s & i = b, \\ \phi_t & i = c, \\ \phi_u & i = d, \\ \vdots & \end{cases} \tag{151}$$

Also, it can be shown that for an N-order Slater determinant, Φ_0^{e},

$$\begin{aligned} \int d\mathbf{r}_{(1)}^{\text{e}} \underset{1i}{\Phi_0^{\text{e}}} \underset{1j}{\Phi_a^r} &= \begin{cases} 0 & i \neq j, \\ (N-1)! & i = j = a. \end{cases} \\ &= (N-1)!\delta_{ij}\delta_{ia} \end{aligned} \tag{152}$$

Now, we will examine each term in equation (150).

A.4.1.1 Term 1.

$$\left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| -\sum_{kk'} r_{kk'}^{-1} \middle| \Phi_a^r \Phi_0^{\mathrm{p}} \right\rangle \tag{153}$$

$$= \left\langle \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \middle| -2\sum_{k}^{N} r_{k1'}^{-1} \middle| \Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \right\rangle \tag{154}$$

$$= -2\sum_{k}^{N} \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) r_{k1'}^{-1} \Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \tag{155}$$

$$= -2N \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \tag{156}$$

$$= -\frac{2N}{N!}\sum_{ij}^{N}(-1)^{i+j} \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1) \Phi_{\substack{0\\1i}}^{\mathrm{e}} \phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \psi_j^r(\mathbf{r}_1) \Phi_{\substack{a\\1j}}^r \phi_{1'}(\mathbf{r}_{1'}) \tag{157}$$

$$\begin{aligned} = &-\frac{2}{(N-1)!}\sum_{ij}^{N}(-1)^{i+j} \int d\mathbf{r}_{(1)}^{\mathrm{e}} \Phi_{\substack{0\\1i}}^{\mathrm{e}} \Phi_{\substack{a\\1j}}^r \\ &\times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \psi_j^r(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) \end{aligned} \tag{158}$$

$$= -\frac{2(N-1)!}{(N-1)!}\sum_{ij}^{N}(-1)^{i+j}\delta_{ij}\delta_{ia} \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \psi_j^r(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) \tag{159}$$

$$= -2\sum_{i}^{N} \cancel{(-1)^{2i}}^{1} \delta_{ia} \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \psi_i^r(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) \tag{160}$$

$$= -2 \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_a(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \phi_r(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) \tag{161}$$

$$= -2\langle a1'|r1'\rangle \tag{162}$$

Here, and in subsequent sections, the electron-positron attraction integrals such as $\langle a1'|r1'\rangle$ are in the MO basis.

A.4.1.2 Term 2.

$$\left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| -\sum_k \nu_{\mathrm{ep,e}}^{\mathrm{HF}}(k) \middle| \Phi_a^r\Phi_0^{\mathrm{p}} \right\rangle \tag{163}$$

$$= \left\langle \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \middle| -2\sum_k^N \nu_{\mathrm{ep,e}}^{\mathrm{HF}}(k) \middle| \Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \right\rangle$$

$$= -2\sum_k^N \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'})\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(k)\Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \tag{164}$$

$$= -2N \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'})\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(1)\Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \tag{165}$$

$$= -\frac{2N}{N!}\sum_{ij}^N (-1)^{i+j} \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\Phi_{\substack{0\\1i}}^{\mathrm{e}}\phi_{1'}(\mathbf{r}_{1'})\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(1)\psi_j^r(\mathbf{r}_1)\Phi_{\substack{a\\1j}}^r\phi_{1'}(\mathbf{r}_{1'}) \tag{166}$$

$$= -\frac{2}{(N-1)!}\sum_{ij}^N (-1)^{i+j} \int d\mathbf{r}_{(1)}^{\mathrm{e}} \Phi_{\substack{0\\1i}}^{\mathrm{e}} \Phi_{\substack{a\\1j}}^r \tag{167}$$

$$\times \cancelto{1}{\int d\mathbf{r}_{1'}\phi_{1'}(\mathbf{r}_{1'})\phi_{1'}(\mathbf{r}_{1'})} \int d\mathbf{r}_1 \phi_i(\mathbf{r}_1)\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(1)\psi_j^r(\mathbf{r}_1)$$

$$= -\frac{2(N-1)!}{(N-1)!}\sum_{ij}^N (-1)^{i+j}\delta_{ij}\delta_{ia} \int d\mathbf{r}_1 \phi_i(\mathbf{r}_1)\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(1)\psi_j^r(\mathbf{r}_1) \tag{168}$$

$$= -2\sum_i^N (-1)^{2i}\delta_{ia} \int d\mathbf{r}_1 \phi_i(\mathbf{r}_1)\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(1)\psi_i^r(\mathbf{r}_1) \tag{169}$$

$$= -2\int d\mathbf{r}_1 \phi_a(\mathbf{r}_1)\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(1)\phi_r(\mathbf{r}_1) \tag{170}$$

$$= -2\int d\mathbf{r}_1 \phi_a(\mathbf{r}_1)\left(-\int d\mathbf{r}_{1'}\phi_{1'}(\mathbf{r}_{1'})r_{11'}^{-1}\phi_{1'}(\mathbf{r}_{1'})\right)\phi_r(\mathbf{r}_1) \tag{171}$$

$$= 2\int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_a(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'})r_{11'}^{-1}\phi_r(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) \tag{172}$$

$$= 2\left\langle a1' \middle| r1' \right\rangle \tag{173}$$

A.4.1.3 Term 3.

$$\begin{aligned}
&\left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| -\sum_{k'} \nu_{\mathrm{ep,p}}^{\mathrm{HF}}(k') \middle| \Phi_a^r\Phi_0^{\mathrm{p}} \right\rangle \\
&\quad = \left\langle \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \middle| -\nu_{\mathrm{ep,p}}^{\mathrm{HF}}(1') \middle| \Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \right\rangle && (174)\\
&\quad = -\int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'})\nu_{\mathrm{ep,p}}^{\mathrm{HF}}(1')\Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) && (175)\\
&\quad = -\int d\mathbf{r}^{\mathrm{e}} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\Phi_a^r(\mathbf{r}^{\mathrm{e}}) \overbrace{\int d\mathbf{r}_{1'}\phi_{1'}(\mathbf{r}_{1'})\nu_{\mathrm{ep,p}}^{\mathrm{HF}}(1')\phi_{1'}(\mathbf{r}_{1'})}^{0} && (176)\\
&\quad = 0 && (177)
\end{aligned}$$

Therefore, terms 1 and 2 cancel to give zero for the matrix element between the HF ground state and the single-electron excitation wavefunction:

$$\left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| \hat{W}_{\mathrm{ep}} \middle| \Phi_a^r\Phi_0^{\mathrm{p}} \right\rangle = -2\left\langle a1' \middle| r1' \right\rangle + 2\left\langle a1' \middle| r1' \right\rangle = 0 \tag{178}$$

A similar result is obtained for the single-positron excitation term.

A.4.2 Single-Electron/Single-Positron Excitation. Next, we compute the matrix element between the NEO-HF ground state wavefunction and the wavefunction consisting of single-electronic and single-positronic excitations:

$$\left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| \hat{W}_{\mathrm{ep}} \middle| \Phi_a^r\Phi_{a'}^{r'} \right\rangle = \left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| \overbrace{-\sum_{kk'} r_{kk'}^{-1}}^{\text{Term 1}} \overbrace{-\sum_{k} \nu_{\mathrm{ep,e}}^{\mathrm{HF}}(k)}^{\text{Term 2}} \overbrace{-\sum_{k'} \nu_{\mathrm{ep,p}}^{\mathrm{HF}}(k')}^{\text{Term 3}} \middle| \Phi_a^r\Phi_{a'}^{r'} \right\rangle \tag{179}$$

Again, we look at each term individually.

A.4.2.1 Term 1.

$$\left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| -\sum_{kk'} r_{kk'}^{-1} \middle| \Phi_a^r \Phi_{a'}^{r'} \right\rangle \tag{180}$$

$$= \left\langle \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \middle| -2\sum_{k}^{N} r_{k1'}^{-1} \middle| \Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{r'}(\mathbf{r}_{1'}) \right\rangle \tag{181}$$

$$= -2\sum_{k}^{N} \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) r_{k1'}^{-1} \Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{r'}(\mathbf{r}_{1'}) \tag{182}$$

$$= -2N \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{r'}(\mathbf{r}_{1'}) \tag{183}$$

$$= -\frac{2N}{N!}\sum_{ij}^{N} (-1)^{i+j} \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1) \Phi^{\mathrm{e}}_{\substack{0 \\ 1i}} \phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \psi^r_{\substack{a \\ j}}(\mathbf{r}_1) \Phi^r_{\substack{a \\ 1j}} \phi_{r'}(\mathbf{r}_{1'}) \tag{184}$$

$$\begin{aligned} = &-\frac{2}{(N-1)!}\sum_{ij}^{N} (-1)^{i+j} \int d\mathbf{r}^{\mathrm{e}}_{(1)} \Phi^{\mathrm{e}}_{\substack{0 \\ 1i}} \Phi^r_{\substack{a \\ 1j}} \\ &\times \int d\mathbf{x}_1 d\mathbf{x}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \psi^r_{\substack{a \\ j}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_{1'}) \end{aligned} \tag{185}$$

$$= -\frac{2(N-1)!}{(N-1)!}\sum_{ij}^{N} (-1)^{i+j} \delta_{ij}\delta_{ia} \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \psi^r_{\substack{a \\ j}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_{1'}) \tag{186}$$

$$= -2\sum_{i}^{N} \cancel{(-1)^{2i}}^{1} \delta_{ia} \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \psi^r_{\substack{a \\ i}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_{1'}) \tag{187}$$

$$= -2 \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_a(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'}) r_{11'}^{-1} \phi_r(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_{1'}) \tag{188}$$

$$= -2\left\langle a1' \middle| rr' \right\rangle \tag{189}$$

A.4.2.2 Term 2.

$$\left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| -\sum_k \nu_{\mathrm{ep,e}}^{\mathrm{HF}}(k) \middle| \Phi_a^r \Phi_{a'}^{r'} \right\rangle$$

$$= \left\langle \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) \middle| -2\sum_k^N \nu_{\mathrm{ep,e}}^{\mathrm{HF}}(k) \middle| \Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{r'}(\mathbf{r}_{1'}) \right\rangle \tag{190}$$

$$= -2\sum_k^N \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'})\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(k)\Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{r'}(\mathbf{r}_{1'}) \tag{191}$$

$$= -2N \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'})\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(1)\Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{r'}(\mathbf{r}_{1'}) \tag{192}$$

$$= -\frac{2N}{N!}\sum_{ij}^N (-1)^{i+j} \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\Phi_{0\,1i}^{\mathrm{e}}\phi_{1'}(\mathbf{r}_{1'})\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(1)\psi_j^r(\mathbf{r}_1)\Phi_{a\,1j}^r\phi_{r'}(\mathbf{r}_{1'}) \tag{193}$$

$$\begin{aligned} &= -\frac{2}{(N-1)!}\sum_{ij}^N (-1)^{i+j} \int d\mathbf{r}_{(1)}^{\mathrm{e}} \Phi_{0\,1i}^{\mathrm{e}} \Phi_{a\,1j}^r \\ &\quad \times \cancelto{0}{\int d\mathbf{r}_{1'}\phi_{1'}(\mathbf{r}_{1'})\phi_{r'}(\mathbf{r}_{1'})} \int d\mathbf{r}_1 \phi_i(\mathbf{r}_1)\nu_{\mathrm{ep,e}}^{\mathrm{HF}}(1)\psi_j^r(\mathbf{r}_1) \end{aligned} \tag{194}$$

$$= 0 \tag{195}$$

A similar result is found for term 3. Therefore, the only nonzero matrix element for the NEO-MP2 electron-positron correlation correction results from the electron-positron attraction energy between the HF ground state wavefunction and the single-electronic/single-positronic excitation:

$$\left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| \hat{W}_{\mathrm{ep}} \middle| \Phi_a^r \Phi_{a'}^{r'} \right\rangle = -2\left\langle a1' \middle| rr' \right\rangle. \tag{196}$$

A.5 Calculation of the Spin-Averaged Positron-Electron Annihilation Rate from the NEO-HF Wavefunction

Now that we have laid some theoretical and notational groundwork for NEO-HF and NEO-MP2, we are ready to develop the equations for computing the positron-electron annihilation observables. First, we will examine the NEO-HF annihilation rate. In general for an N_e-electron, N_p-positron system, the spin-averaged annihilation rate is given by

$$\lambda = 4\pi r_0^2 c \sum_{i=1}^{N_e} \sum_{j=1}^{N_p} \int \Psi^\dagger(\mathbf{x}^e, \mathbf{x}^p)\, \hat{O}_{i,j}^{S\dagger}\, \delta(\mathbf{r}_i^e - \mathbf{r}_j^p)\, \hat{O}_{i,j}^{S}\, \Psi(\mathbf{x}^e, \mathbf{x}^p) d\mathbf{x}^e d\mathbf{x}^p \tag{197}$$

where $\delta(\mathbf{r}_i^e - \mathbf{r}_j^p)$ is the Dirac delta function for the ith electron and the jth positron. [38] The operator $\hat{O}_{i,j}^S$ is a spin projection operator to the singlet state of the ith electron-jth positron pair, which can be written as

$$\hat{O}_{i,j}^S = \left(1 - \frac{1}{2} S_{i,j}^2\right). \tag{198}$$

For an electronically closed-shell system with N doubly-occupied electron orbitals and a single positron, equation (197) simplifies to:

$$\lambda = 2\pi r_0^2 c \sum_{i=1}^{N} \int_{\mathbb{R}^{3(N+1)}} \Psi^\dagger(\mathbf{r}^e, \mathbf{r}^p) \delta(\mathbf{r}_i^e - \mathbf{r}^p) \Psi(\mathbf{r}^e, \mathbf{r}^p) d\mathbf{r}^e d\mathbf{r}^p \tag{199}$$

A.5.1 Example: e^+LiH . Let's look at the simplest non-trivial bound system consisting of an electronic closed-shell: e^+LiH. For this case,

$$\begin{aligned} \Phi_0^e &= \frac{1}{\sqrt{2}} \begin{vmatrix} \phi_1^e(\mathbf{r}_1^e) & \phi_2^e(\mathbf{r}_1^e) \\ \phi_1^e(\mathbf{r}_2^e) & \phi_2^e(\mathbf{r}_2^e) \end{vmatrix} \\ &= \frac{1}{\sqrt{2}} \left[\phi_1^e(\mathbf{r}_1^e)\phi_2^e(\mathbf{r}_2^e) - \phi_1^e(\mathbf{r}_2^e)\phi_2^e(\mathbf{r}_1^e)\right], \end{aligned} \tag{200}$$

and the total NEO-HF wavefunction is given by

$$\Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} = \frac{1}{\sqrt{2}}\left[\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}}) - \phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\right]\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \tag{201}$$

Substituting the electronic and positronic wave functions into equation (199) (and dropping the complex conjugate notation since our spatial orbitals are pure real functions),

$$\begin{aligned}
\lambda = 2\pi r_0^2 c \sum_{i=1}^{2} \int \Bigg\{ & \frac{1}{\sqrt{2}}\left[\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}}) - \phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\right]\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \\
& \times \delta(\mathbf{r}_i^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) \\
& \times \frac{1}{\sqrt{2}}\left[\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}}) - \phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\right]\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \Bigg\} d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}_2^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}}
\end{aligned} \tag{202}$$

$$\begin{aligned}
= \pi r_0^2 c \int \sum_{i=1}^{2} \Big\{ & \left[\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) - \phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right] \\
& \times \left[\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) - \phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right] \\
& \times \delta(\mathbf{r}_i^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) \Big\} d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}_2^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}}
\end{aligned} \tag{203}$$

$$\begin{aligned}
= \pi r_0^2 c \int \sum_{i=1}^{2} \Bigg\{ & \Bigg[\Big(\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\Big)^2 \\
& - 2\Big(\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\Big)\Big(\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\Big) \\
& + \Big(\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}_1^{\mathrm{p}})\Big)^2 \Bigg] \delta(\mathbf{r}_i^{\mathrm{e}} - \mathbf{r}_1^{\mathrm{p}}) \Bigg\} d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}_2^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}}
\end{aligned} \tag{204}$$

Next, we can collect terms and we are left with three integrals involving a δ-function between $\mathbf{r}_1^{\mathrm{e}}$ and $\mathbf{r}^{\mathrm{p}}$ and three with the δ-function between $\mathbf{r}_2^{\mathrm{e}}$ and $\mathbf{r}^{\mathrm{p}}$.

$$\begin{aligned}\lambda = \pi r_0^2 c \int \Bigg\{ \Bigg[& \overbrace{\left(\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right)^2}^{\text{term 1}} \\ & \overbrace{-2\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})}^{\text{term 2}} \\ & + \overbrace{\left(\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right)^2}^{\text{term 3}} \Bigg] \delta(\mathbf{r}_1^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) \\ & + \Bigg[\overbrace{\left(\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right)^2}^{\text{term 4}} \\ & \overbrace{-2\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})}^{\text{term 5}} \\ & + \overbrace{\left(\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right)^2}^{\text{term 6}} \Bigg] \delta(\mathbf{r}_2^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) \Bigg\} d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}_2^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}}\end{aligned} \tag{205}$$

A.5.1.1 Term 1.

$$\begin{aligned}\int \left[\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2 \delta(\mathbf{r}_1^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}_2^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}} \\ = \int \left[\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\right]^2 d\mathbf{r}_2^{\mathrm{e}} \iint \left[\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2 \delta(\mathbf{r}_1^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}} \\ = \cancelto{1}{\int \left[\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\right]^2 d\mathbf{r}_2^{\mathrm{e}}} \int \left[\phi_1^{\mathrm{e}}(\mathbf{r}^{\mathrm{p}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2 d\mathbf{r}^{\mathrm{p}} = S_{11}^{11},\end{aligned} \tag{206}$$

where the orthonormal nature of the spatial orbitals,

$$\int \phi_i^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_i^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}}) d\mathbf{r}_1^{\mathrm{e}} = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases} \tag{207}$$

was used. Also, notice we have introduced the "four-overlap integral" notation:

$$S_{ij}^{kl} = \int \phi_i^{\mathrm{e}} \phi_j^{\mathrm{e}} \phi_k^{\mathrm{p}} \phi_l^{\mathrm{p}} \, , \tag{208}$$

which is the integral of the product of 4 spatial orbitals.

A.5.1.2 Term 2.

$$\begin{aligned} &- \int 2\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\delta(\mathbf{r}_1^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}_2^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}} \\ &\quad = -\cancelto{0}{\int 2\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}}) d\mathbf{r}_2^{\mathrm{e}}} \iint \phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\delta(\mathbf{r}_1^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}}, \end{aligned} \tag{209}$$

where we have used the orthogonality of ϕ_1^{e} and ϕ_2^{e}.

A.5.1.3 Term 3.

$$\begin{aligned} &\int \left[\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2 \delta(\mathbf{r}_1^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}_2^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}} \\ &\qquad = \int \left[\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\right]^2 d\mathbf{r}_2^{\mathrm{e}} \iint \left[\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2 \delta(\mathbf{r}_1^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}} \\ &\qquad\qquad = \cancelto{1}{\int \left[\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\right]^2 d\mathbf{r}_2^{\mathrm{e}}} \int \left[\phi_2^{\mathrm{e}}(\mathbf{r}^{\mathrm{p}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2 d\mathbf{r}^{\mathrm{p}} = S_{22}^{11} \end{aligned} \tag{210}$$

A.5.1.4 Term 4.

$$\begin{aligned} &\int \left[\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2 \delta(\mathbf{r}_2^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) d\mathbf{r}_1^{\mathrm{e}} d\mathbf{r}_2^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}} \\ &\qquad = \cancelto{1}{\int \left[\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\right]^2 d\mathbf{r}_1^{\mathrm{e}}} \iint \left[\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2 \delta(\mathbf{r}_2^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) d\mathbf{r}_2^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}} \\ &\qquad\qquad = \int \left[\phi_2^{\mathrm{e}}(\mathbf{r}^{\mathrm{p}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2 d\mathbf{r}^{\mathrm{p}} = S_{22}^{11} \end{aligned}$$

A.5.1.5 Term 5.

$$-2\int \phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\delta(\mathbf{r}_2^{\mathrm{e}}-\mathbf{r}^{\mathrm{p}})d\mathbf{r}_1^{\mathrm{e}}d\mathbf{r}_2^{\mathrm{e}}d\mathbf{r}^{\mathrm{p}}$$
$$= -2\cancelto{0}{\int \phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})d\mathbf{r}_1^{\mathrm{e}}}\iint \phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\phi_2^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\delta(\mathbf{r}_2^{\mathrm{e}}-\mathbf{r}^{\mathrm{p}})d\mathbf{r}_2^{\mathrm{e}}d\mathbf{r}^{\mathrm{p}} = 0 \quad (211)$$

A.5.1.6 Term 6.

$$\int \left[\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2\delta(\mathbf{r}_2^{\mathrm{e}}-\mathbf{r}^{\mathrm{p}})d\mathbf{r}_1^{\mathrm{e}}d\mathbf{r}_2^{\mathrm{e}}d\mathbf{r}^{\mathrm{p}}$$
$$= \cancelto{1}{\int [\phi_2^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})]^2 d\mathbf{r}_1^{\mathrm{e}}}\iint \left[\phi_1^{\mathrm{e}}(\mathbf{r}_2^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2\delta(\mathbf{r}_2^{\mathrm{e}}-\mathbf{r}^{\mathrm{p}})d\mathbf{r}_2^{\mathrm{e}}d\mathbf{r}^{\mathrm{p}}$$
$$= \int \left[\phi_1^{\mathrm{e}}(\mathbf{r}^{\mathrm{p}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\right]^2 d\mathbf{r}^{\mathrm{p}} = S_{11}^{11}$$

Combining all terms, we find that

$$\lambda = 2\pi {r_0}^2 c\left[S_{11}^{11}+S_{22}^{11}\right] \quad (212)$$

A.5.2 Annihilation Rate for an Arbitrary NEO-HF Wavefunction. Next, we will investigate the annihilation rate for an arbitrary NEO-HF wavefunction. Substituting the general form of our wavefunction into equation (199), we get:

$$\lambda = 2\pi r_0^2 c\int_{\mathbb{R}^{3(N+1)}} \Phi_0^{\mathrm{e}*}(\mathbf{r}^{\mathrm{e}})\phi_1^{\mathrm{p}*}(\mathbf{r}^{\mathrm{p}})\left(\sum_{k=1}^{N}\delta(\mathbf{r}_k^{\mathrm{e}}-\mathbf{r}^{\mathrm{p}})\right)\Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})d\mathbf{r}^{\mathrm{e}}d\mathbf{r}^{\mathrm{p}} \quad (213)$$

We can drop the complex conjugate notation since our spatial orbitals are real functions, and the expression can be rewritten using the expansion in minors:

$$\lambda = \frac{2\pi r_0^2 c}{N!}\int_{\mathbb{R}^{3(N+1)}}\sum_{i=1}^{N}(-1)^{i+1}\phi_i^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\Phi_{0\ 1i}^{\mathrm{e}}\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})\sum_{k=1}^{N}\delta(\mathbf{r}_k^{\mathrm{e}}-\mathbf{r}^{\mathrm{p}})$$
$$\times\sum_{j=1}^{N}(-1)^{j+1}\phi_j^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\Phi_{0\ 1j}^{\mathrm{e}}\phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}})d\mathbf{r}^{\mathrm{e}}d\mathbf{r}^{\mathrm{p}} \quad (214)$$

Combining like terms, we see that

$$\lambda = \frac{2\pi r_0^2 c}{N!} \int_{\mathbb{R}^{3(N+1)}} \sum_{i,j=1}^{N} (-1)^{i+j} \phi_i^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}}) \phi_j^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}}) \phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \Phi^{\mathrm{e}}_{\substack{0 \\ 1i}} \Phi^{\mathrm{e}}_{\substack{0 \\ 1j}} \times \sum_{k=1}^{N} \delta(\mathbf{r}_k^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) d\mathbf{r}^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}} \tag{215}$$

We are free to expand our determinants over any row and any column. Therefore, instead of expanding over $1i$ and $1j$, we choose to expand the determinant in the kth sum over the kth column. Since the δ-function involves the kth electron, we will soon see that this is a convenient choice.

$$\begin{aligned} \lambda &= \frac{2\pi r_0^2 c}{N!} \sum_{i,j,k=1}^{N} (-1)^{i+j} \int_{\mathbb{R}^{3(N+1)}} \phi_i^{\mathrm{e}}(\mathbf{r}_k^{\mathrm{e}}) \phi_j^{\mathrm{e}}(\mathbf{r}_k^{\mathrm{e}}) \phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \Phi^{\mathrm{e}}_{\substack{0 \\ ki}} \Phi^{\mathrm{e}}_{\substack{0 \\ kj}} \delta(\mathbf{r}_k^{\mathrm{e}} - \mathbf{r}^{\mathrm{p}}) d\mathbf{r}^{\mathrm{e}} d\mathbf{r}^{\mathrm{p}} \\ &= \frac{2\pi r_0^2 c}{N!} \sum_{i,j,k=1}^{N} (-1)^{i+j} \int_{\mathbb{R}^{3N}} \phi_i^{\mathrm{e}}(\mathbf{r}^{\mathrm{p}}) \phi_j^{\mathrm{e}}(\mathbf{r}^{\mathrm{p}}) \phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \Phi^{\mathrm{e}}_{\substack{0 \\ ki}} \Phi^{\mathrm{e}}_{\substack{0 \\ kj}} d\mathbf{r}^{\mathrm{e}}_{(k)} d\mathbf{r}^{\mathrm{p}} \end{aligned} \tag{216}$$

Here we integrated over coordinates of the kth electron which accounts for the δ-function and replaces the coordinate of the kth electron with that of the positron. Continuing, we see that

$$\begin{aligned} \lambda &= \frac{2\pi r_0^2 c}{N!} \sum_{i,j=1}^{N} (-1)^{i+j} \int_{\mathbb{R}^3} \phi_i^{\mathrm{e}}(\mathbf{r}^{\mathrm{p}}) \phi_j^{\mathrm{e}}(\mathbf{r}^{\mathrm{p}}) \phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) \phi_1^{\mathrm{p}}(\mathbf{r}^{\mathrm{p}}) d\mathbf{r}^{\mathrm{p}} \sum_{k=1}^{N} \int_{\mathbb{R}^{3(N-1)}} \Phi^{\mathrm{e}}_{\substack{0 \\ ki}} \Phi^{\mathrm{e}}_{\substack{0 \\ kj}} d\mathbf{r}^{\mathrm{e}}_{(k)} \\ &= \frac{2\pi r_0^2 c}{N!} \sum_{i,j=1}^{N} (-1)^{i+j} S_{ij}^{11} \sum_{k=1}^{N} \int_{\mathbb{R}^3(N-1)} \Phi^{\mathrm{e}}_{\substack{0 \\ ki}} \Phi^{\mathrm{e}}_{\substack{0 \\ kj}} d\mathbf{r}^{\mathrm{e}}_{(k)} \end{aligned} \tag{217}$$

Thus, the annihilation rate of an N-electron, one-positron NEO-HF wavefunction reduces to the product of a 4-overlap integral and an interdeterminant overlap integral. Substituting the result of equation (134), the HF annihilation rate reduces to a simple sum over four-

overlap integrals:

$$\lambda = \frac{2\pi r_0^2 c}{N!} \sum_{i=1}^{N} \sum_{j=1}^{N} (-1)^{i+j} S_{ij}^{11} N! \delta_{ij} \tag{218}$$

$$= 2\pi r_0^2 c \sum_{i=1}^{N} \cancel{(-1)^{2i}}^{1} S_{ii}^{11} \tag{219}$$

$$= 2\pi r_0^2 c \sum_{i=1}^{N} S_{ii}^{11} \tag{220}$$

Now let's re-examine our simple example, e^+LiH, with this general equation.

$$\lambda = 2\pi r_0^2 c \sum_{i=1}^{2} S_{ii}^{11} \tag{221}$$

$$= 2\pi r_0^2 c \left[S_{11}^{11} + S_{22}^{11} \right] \tag{222}$$

This matches our previous result.

A.6 Annihilation Rate NEO-MP2 First-Order Correction

For a NEO-HF wavefunction, the first-order perturbation correction to the annihilation rate is given by

$$\lambda^{(1)} = \left\langle \Psi^{(0)} \middle| 2\pi r_0^2 c \sum_i^N \delta(r_i - r_{1'}) \middle| \Psi^{(1)} \right\rangle \tag{223}$$

$$= 2N\pi r_0^2 c \left\langle \Psi^{(0)} \middle| \delta(r_1 - r_{1'}) \middle| \Psi^{(1)} \right\rangle \tag{224}$$

$$= 2N\pi r_0^2 c \left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| \delta(r_1 - r_{1'}) \middle| \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C_{ab}^{rs}\Phi_{ab}^{rs}\Phi_0^{\mathrm{p}} - \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'}\Phi_a^r\Phi_{1'}^{r'} \right\rangle \tag{225}$$

$$= \lambda_{\mathrm{ee}}^{(1)} + \lambda_{\mathrm{ep}}^{(1)}, \tag{226}$$

where

$$\lambda_{\mathrm{ee}}^{(1)} = 2N\pi r_0^2 c \left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| \delta(r_1 - r_{1'}) \middle| \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C_{ab}^{rs}\Phi_{ab}^{rs}\Phi_0^{\mathrm{p}} \right\rangle \tag{227}$$

and

$$\lambda_{\mathrm{ep}}^{(1)} = -2N\pi r_0^2 c \left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| \delta(r_1 - r_{1'}) \middle| \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'}\Phi_a^r\Phi_{1'}^{r'} \right\rangle \tag{228}$$

We will examine each of these term individually, but first we need to introduce a new type of expansion by minors for an N-order Slater determinant, Φ_0^{e}, in order to account for double-electronic excitations:

$$\int d\mathbf{r}_{(1,2)}^{\mathrm{e}} \Phi_{0\ 1i2j}^{\mathrm{e}} \Phi_{ab\ 1k2l}^{rs} = \begin{cases} 0 & i \neq j, \\ (N-2)! & i = k = a \text{ and } j = l = b. \end{cases}$$

$$= (N-2)!\delta_{ik}\delta_{ia}\delta_{jl}\delta_{jb} \tag{229}$$

A.6.1 $\lambda_{ee}^{(1)}$ Term.

$$\lambda_{\rm ee}^{(1)} = 2N\pi r_0^2 c \left\langle \Phi_0^{\rm e}\Phi_0^{\rm p} \middle| \delta(r_1 - r_{1'}) \middle| \sum_{\substack{a<b \\ r<s}} {}^{\rm ee}C_{ab}^{rs}\Phi_{ab}^{rs}\Phi_0^{\rm p} \right\rangle \tag{230}$$

$$= 2N\pi r_0^2 c \sum_{\substack{a<b \\ r<s}} {}^{\rm ee}C_{ab}^{rs} \int d\mathbf{r}^{\rm e} d\mathbf{r}_{1'} \Phi_0^{\rm e}(\mathbf{r}^{\rm e})\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\Phi_{ab}^{rs}(\mathbf{r}^{\rm e})\phi_{1'}(\mathbf{r}_{1'}) \tag{231}$$

$$\begin{aligned} &= \frac{2N\pi r_0^2 c}{N!} \sum_{\substack{a<b \\ r<s}} {}^{\rm ee}C_{ab}^{rs} \sum_{i<j}\sum_{k<l} (-1)^{i+j+1}(-1)^{k+l+1} \\ &\quad \times \int d\mathbf{r}^{\rm e} d\mathbf{r}_{1'} \begin{vmatrix} \phi_i(\mathbf{r}_1) & \phi_i(\mathbf{r}_2) \\ \phi_j(\mathbf{r}_1) & \phi_j(\mathbf{r}_2) \end{vmatrix} \Phi_{0\ 1i2j}^{\rm e}\ \phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'}) \\ &\qquad \times \begin{vmatrix} \psi_{ab\ k}^{rs}(\mathbf{r}_1) & \psi_{ab\ k}^{rs}(\mathbf{r}_2) \\ \psi_{ab\ l}^{rs}(\mathbf{r}_1) & \psi_{ab\ l}^{rs}(\mathbf{r}_2) \end{vmatrix} \Phi_{ab\ 1k2l}^{rs}\ \phi_{1'}(\mathbf{r}_{1'}) \end{aligned} \tag{232}$$

$$\begin{aligned} &= \frac{2\pi r_0^2 c}{(N-1)!} \sum_{\substack{a<b \\ r<s}} {}^{\rm ee}C_{ab}^{rs} \sum_{i<j}\sum_{k<l} (-1)^{i+j+k+l} \\ &\quad \times \int d\mathbf{r}^{\rm e}_{(1,2)} \Phi_{0\ 1i2j}^{\rm e}\ \Phi_{ab\ 1k2l}^{rs} \\ &\quad \times \int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_{1'} \begin{vmatrix} \phi_i(\mathbf{r}_1) & \phi_i(\mathbf{r}_2) \\ \phi_j(\mathbf{r}_1) & \phi_j(\mathbf{r}_2) \end{vmatrix} \phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'}) \\ &\qquad \times \begin{vmatrix} \psi_{ab\ k}^{rs}(\mathbf{r}_1) & \psi_{ab\ k}^{rs}(\mathbf{r}_2) \\ \psi_{ab\ l}^{rs}(\mathbf{r}_1) & \psi_{ab\ l}^{rs}(\mathbf{r}_2) \end{vmatrix} \phi_{1'}(\mathbf{r}_{1'}) \end{aligned} \tag{233}$$

$$\begin{aligned} &= \frac{2(N-2)!\pi r_0^2 c}{(N-1)!} \sum_{\substack{a<b \\ r<s}} {}^{\rm ee}C_{ab}^{rs} \sum_{i<j}\sum_{k<l} (-1)^{i+j+k+l}\delta_{ik}\delta_{ia}\delta_{jl}\delta_{jb} \\ &\quad \times \int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_{1'} \begin{vmatrix} \phi_i(\mathbf{r}_1) & \phi_i(\mathbf{r}_2) \\ \phi_j(\mathbf{r}_1) & \phi_j(\mathbf{r}_2) \end{vmatrix} \phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'}) \\ &\qquad \times \begin{vmatrix} \psi_{ab\ k}^{rs}(\mathbf{r}_1) & \psi_{ab\ k}^{rs}(\mathbf{r}_2) \\ \psi_{ab\ l}^{rs}(\mathbf{r}_1) & \psi_{ab\ l}^{rs}(\mathbf{r}_2) \end{vmatrix} \phi_{1'}(\mathbf{r}_{1'}) \end{aligned} \tag{234}$$

$$
\begin{aligned}
=&\frac{2\pi r_0^2 c}{N-1}\sum_{\substack{a<b\\ r<s}}{}^{\mathrm{ee}}C_{ab}^{rs}\sum_{i<j}\cancel{(-1)^{2i+2j}}^{1}\delta_{ia}\delta_{jb} \\
&\times\int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_{1'}\begin{vmatrix}\phi_i(\mathbf{r}_1) & \phi_i(\mathbf{r}_2)\\ \phi_j(\mathbf{r}_1) & \phi_j(\mathbf{r}_2)\end{vmatrix}\phi_{1'}(\mathbf{r}_{1'})\delta(r_1-r_{1'}) \\
&\quad\times\begin{vmatrix}\underset{i}{\psi_{ab}^{rs}}(\mathbf{r}_1) & \underset{i}{\psi_{ab}^{rs}}(\mathbf{r}_2)\\ \underset{j}{\psi_{ab}^{rs}}(\mathbf{r}_1) & \underset{j}{\psi_{ab}^{rs}}(\mathbf{r}_2)\end{vmatrix}\phi_{1'}(\mathbf{r}_{1'})
\end{aligned}
\tag{235}
$$

$$
\begin{aligned}
=&\frac{2\pi r_0^2 c}{N-1}\sum_{\substack{a<b\\ r<s}}{}^{\mathrm{ee}}C_{ab}^{rs}\int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_{1'}\begin{vmatrix}\phi_a(\mathbf{r}_1) & \phi_a(\mathbf{r}_2)\\ \phi_b(\mathbf{r}_1) & \phi_b(\mathbf{r}_2)\end{vmatrix}\phi_{1'}(\mathbf{r}_{1'})\delta(r_1-r_{1'}) \\
&\quad\times\begin{vmatrix}\phi_r(\mathbf{r}_1) & \phi_r(\mathbf{r}_2)\\ \phi_s(\mathbf{r}_1) & \phi_s(\mathbf{r}_2)\end{vmatrix}\phi_{1'}(\mathbf{r}_{1'})
\end{aligned}
\tag{236}
$$

$$
\begin{aligned}
=&\frac{2\pi r_0^2 c}{N-1}\sum_{\substack{a<b\\ r<s}}{}^{\mathrm{ee}}C_{ab}^{rs}\int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_{1'} \\
&\quad\times\left[\phi_a(\mathbf{r}_1)\phi_b(\mathbf{r}_2)-\phi_a(\mathbf{r}_2)\phi_b(\mathbf{r}_1)\right]\phi_{1'}(\mathbf{r}_{1'})\delta(r_1-r_{1'}) \\
&\quad\times\left[\phi_r(\mathbf{r}_1)\phi_s(\mathbf{r}_2)-\phi_r(\mathbf{r}_2)\phi_s(\mathbf{r}_1)\right]\phi_{1'}(\mathbf{r}_{1'})
\end{aligned}
\tag{237}
$$

$$
\begin{aligned}
=&\frac{2\pi r_0^2 c}{N-1}\sum_{\substack{a<b\\ r<s}}{}^{\mathrm{ee}}C_{ab}^{rs}\int d\mathbf{r}_1 d\mathbf{r}_2 d\mathbf{r}_{1'}\phi_{1'}(\mathbf{r}_{1'})\delta(r_1-r_{1'})\phi_{1'}(\mathbf{r}_{1'}) \\
&\quad\times\Big\{\left[\phi_a(\mathbf{r}_1)\phi_b(\mathbf{r}_2)\phi_r(\mathbf{r}_1)\phi_s(\mathbf{r}_2)\right] \\
&\qquad-\left[\phi_a(\mathbf{r}_1)\phi_b(\mathbf{r}_2)\phi_r(\mathbf{r}_2)\phi_s(\mathbf{r}_1)\right] \\
&\qquad-\left[\phi_a(\mathbf{r}_2)\phi_b(\mathbf{r}_1)\phi_r(\mathbf{r}_1)\phi_s(\mathbf{r}_2)\right] \\
&\qquad+\left[\phi_a(\mathbf{r}_2)\phi_b(\mathbf{r}_1)\phi_r(\mathbf{r}_2)\phi_s(\mathbf{r}_1)\right]\Big\}
\end{aligned}
\tag{238}
$$

$$
\begin{aligned}
&= \frac{2\pi r_0^2 c}{N-1} \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C_{ab}^{rs} \\
&\quad \times \Bigg\{ \cancelto{0}{\int d\mathbf{r}_2 \phi_b(\mathbf{r}_2)\phi_s(\mathbf{r}_2)} \\
&\qquad \times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_a(\mathbf{r}_1)\phi_r(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\phi_{1'}(\mathbf{r}_{1'}) \\
&\quad - \cancelto{0}{\int d\mathbf{r}_2 \phi_b(\mathbf{r}_2)\phi_r(\mathbf{r}_2)} \\
&\qquad \times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_a(\mathbf{r}_1)\phi_s(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\phi_{1'}(\mathbf{r}_{1'}) \\
&\quad - \cancelto{0}{\int d\mathbf{r}_2 \phi_a(\mathbf{r}_2)\phi_s(\mathbf{r}_2)} \\
&\qquad \times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_b(\mathbf{r}_1)\phi_r(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\phi_{1'}(\mathbf{r}_{1'}) \\
&\quad + \cancelto{0}{\int d\mathbf{r}_2 \phi_a(\mathbf{r}_2)\phi_r(\mathbf{r}_2)} \\
&\qquad \times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_b(\mathbf{r}_1)\phi_s(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\phi_{1'}(\mathbf{r}_{1'}) \Bigg\}
\end{aligned}
\tag{239}
$$

$$
= 0 \tag{240}
$$

A.6.2 $\lambda_{ep}^{(1)}$ Term.

$$\lambda_{\mathrm{ep}}^{(1)} = -2N\pi r_0^2 c \left\langle \Phi_0^{\mathrm{e}}\Phi_0^{\mathrm{p}} \middle| \delta(r_1 - r_{1'}) \middle| \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'}\Phi_a^r\Phi_{1'}^{r'} \right\rangle \tag{241}$$

$$= -2N\pi r_0^2 c \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{r'}(\mathbf{r}_{1'}) \tag{242}$$

$$\begin{aligned} = &-\frac{2N}{N!}\pi r_0^2 c \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \sum_{ij}(-1)^{i+j} \\ &\times \int d\mathbf{r}^{\mathrm{e}} d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\Phi^{\mathrm{e}}_{\substack{0\\1i}}\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\psi^r_j(\mathbf{r}_1)\Phi^r_{\substack{a\\1j}}\phi_{r'}(\mathbf{r}_{1'}) \end{aligned} \tag{243}$$

$$\begin{aligned} = &-\frac{2}{(N-1)!}\pi r_0^2 c \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \sum_{ij}(-1)^{i+j} \int d\mathbf{r}^{\mathrm{e}}_{(1)} \Phi^{\mathrm{e}}_{\substack{0\\1i}} \Phi^r_{\substack{a\\1j}} \\ &\times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\psi^r_j(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_{1'}) \end{aligned} \tag{244}$$

$$\begin{aligned} = &-\frac{2(N-1)!}{(N-1)!}\pi r_0^2 c \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \sum_{ij}(-1)^{i+j}\delta_{ij}\delta_{ia} \\ &\times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\psi^r_j(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_{1'}) \end{aligned} \tag{245}$$

$$\begin{aligned} = &-2\pi r_0^2 c \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \sum_{i} \cancel{(-1)^{2i}}^{1}\delta_{ia} \\ &\times \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\psi_i(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_{1'}) \end{aligned} \tag{246}$$

$$= -2\pi r_0^2 c \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \int d\mathbf{r}_1 d\mathbf{r}_{1'} \phi_a(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_{1'})\delta(r_1 - r_{1'})\phi_r(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_{1'}) \tag{247}$$

$$= -2\pi r_0^2 c \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \int d\mathbf{r}_{1'} \phi_a(\mathbf{r}_{1'})\phi_{1'}(\mathbf{r}_{1'})\phi_r(\mathbf{r}_{1'})\phi_{r'}(\mathbf{r}_{1'}) \tag{248}$$

$$= -2\pi r_0^2 c \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} S_{ar}^{1'r'} \tag{249}$$

Therefore, we see that the $\lambda_{\mathrm{ee}}^{(1)}$ term is zero, and as with the energy expressions, the only nonzero first-order perturbation correction to λ results from the single-electronic/single-positronic term.

A.7 Calculation of the Four-Overlap Integral

As seen in previous sections, the calculation of the annihilation rate from the NEO-HF electron-positron wavefunction is simply a matter of calculating the four-overlap integrals, $S_{ii}^{1'1'}$. Also, for the first-order correction to the annihilation rate, it is necessary to calculate four-overlap integrals of the form $S_{ar}^{1'r'}$. In order to compute these integrals, we need to develop the general form of the four-overlap integral across four Gaussian primitives of arbitrary angular momentum, which we will denote with a lower-case s (to distinguish from the four-overlap integral that was previously defined as being over molecular orbitals):

$$s_{12}^{34} = \int_{\mathbb{R}^3} \varphi_1(\mathbf{r})\varphi_2(\mathbf{r})\varphi_3(\mathbf{r})\varphi_4(\mathbf{r})d\mathbf{r} \tag{250}$$

where φ_i is a Gaussian primitive with angular momentum numbers $\{l_i, m_i, n_i\}$ and exponent α_i located at point $\mathbf{A}_i$:

$$\varphi_i(\mathbf{r}) = x_{\mathbf{A}_i}^{l_i} y_{\mathbf{A}_i}^{m_i} z_{\mathbf{A}_i}^{n_i} e^{-\alpha_i (x_{\mathbf{A}_i}^2 + y_{\mathbf{A}_i}^2 + z_{\mathbf{A}_i}^2)} \tag{251}$$

$$= x_{\mathbf{A}_i}^{l_i} y_{\mathbf{A}_i}^{m_i} z_{\mathbf{A}_i}^{n_i} e^{-\alpha_i \mathbf{r}_{\mathbf{A}_i}^2}, \tag{252}$$

with

$$x_{\mathbf{A}_i} = x - \mathbf{A}_i \tag{253}$$

Using the Gaussian Product Theorem,

$$\begin{aligned} s_{12}^{34} = \iiint & x_{\mathbf{A}_1}^{l_1} x_{\mathbf{A}_2}^{l_2} x_{\mathbf{A}_3}^{l_3} x_{\mathbf{A}_4}^{l_4} y_{\mathbf{A}_1}^{m_1} y_{\mathbf{A}_2}^{m_2} y_{\mathbf{A}_3}^{m_3} y_{\mathbf{A}_4}^{m_4} z_{\mathbf{A}_1}^{n_1} z_{\mathbf{A}_2}^{n_2} z_{\mathbf{A}_3}^{n_3} z_{\mathbf{A}_4}^{n_4} \\ & \times \exp\left[-\alpha_1\alpha_2 \left(\overline{\mathbf{A}_1\mathbf{A}_2}\right)^2 / \gamma_p\right] \exp\left[-\alpha_3\alpha_4 \left(\overline{\mathbf{A}_3\mathbf{A}_4}\right)^2 / \gamma_q\right] \\ & \times e^{-\gamma_p \mathbf{r}_{\mathbf{P}}^2} e^{-\gamma_q \mathbf{r}_{\mathbf{Q}}^2}\, dx\, dy\, dz, \end{aligned} \tag{254}$$

where

$$\gamma_p = \alpha_1 + \alpha_2 \qquad \mathbf{P} = \frac{\alpha_1 \mathbf{A}_1 + \alpha_2 \mathbf{A}_2}{\gamma_p} \tag{255}$$

$$\gamma_q = \alpha_3 + \alpha_4 \qquad \mathbf{Q} = \frac{\alpha_3 \mathbf{A}_3 + \alpha_4 \mathbf{A}_4}{\gamma_q} \tag{256}$$

Applying the Gaussian Product Theorem again,

$$\begin{aligned} s_{12}^{34} = & \iiint x_{\mathbf{A}_1}^{l_1} x_{\mathbf{A}_2}^{l_2} x_{\mathbf{A}_3}^{l_3} x_{\mathbf{A}_4}^{l_4} y_{\mathbf{A}_1}^{m_1} y_{\mathbf{A}_2}^{m_2} y_{\mathbf{A}_3}^{m_3} y_{\mathbf{A}_4}^{m_4} z_{\mathbf{A}_1}^{n_1} z_{\mathbf{A}_2}^{n_2} z_{\mathbf{A}_3}^{n_3} z_{\mathbf{A}_4}^{n_4} \\ & \times \exp\left[-\alpha_1\alpha_2 \left(\overline{\mathbf{A}_1\mathbf{A}_2}\right)^2 / \gamma_p\right] \exp\left[-\alpha_3\alpha_4 \left(\overline{\mathbf{A}_3\mathbf{A}_4}\right)^2 / \gamma_q\right] \\ & \times \exp\left[-\gamma_p\gamma_q \left(\overline{\mathbf{PQ}}\right)^2 / \gamma_t\right] e^{-\gamma_t \mathbf{r}_\mathbf{T}^2} \, dx \, dy \, dz \end{aligned} \tag{257}$$

$$\begin{aligned} = & \iiint x_{\mathbf{A}_1}^{l_1} x_{\mathbf{A}_2}^{l_2} x_{\mathbf{A}_3}^{l_3} x_{\mathbf{A}_4}^{l_4} y_{\mathbf{A}_1}^{m_1} y_{\mathbf{A}_2}^{m_2} y_{\mathbf{A}_3}^{m_3} y_{\mathbf{A}_4}^{m_4} z_{\mathbf{A}_1}^{n_1} z_{\mathbf{A}_2}^{n_2} z_{\mathbf{A}_3}^{n_3} z_{\mathbf{A}_4}^{n_4} \\ & \times \exp\left[-\alpha_1\alpha_2 \left(\overline{\mathbf{A}_1\mathbf{A}_2}\right)^2 / \gamma_p\right] \exp\left[-\alpha_3\alpha_4 \left(\overline{\mathbf{A}_3\mathbf{A}_4}\right)^2 / \gamma_q\right] \\ & \times \exp\left[-\gamma_p\gamma_q \left(\overline{\mathbf{PQ}}\right)^2 / \gamma_t\right] e^{-\gamma_t x_\mathbf{T}^2} e^{-\gamma_t y_\mathbf{T}^2} e^{-\gamma_t z_\mathbf{T}^2} \, dx \, dy \, dz, \end{aligned} \tag{258}$$

where

$$\gamma_t = \gamma_p + \gamma_q \tag{259}$$

and

$$\mathbf{T} = \frac{\gamma_p \mathbf{P} + \gamma_q \mathbf{Q}}{\gamma_t} \tag{260}$$

Separating variables, s_{12}^{34} can be rewritten as

$$\begin{aligned} s_{12}^{34} = & \exp[-\alpha_1\alpha_2 \left(\overline{\mathbf{A}_1\mathbf{A}_2}\right)^2 / \gamma_p] \\ & \times \exp[-\alpha_3\alpha_4 \left(\overline{\mathbf{A}_3\mathbf{A}_4}\right)^2 / \gamma_q] \\ & \times \exp[-\gamma_p\gamma_q \left(\overline{\mathbf{PQ}}\right)^2 / \gamma_t] \\ & \times I_x I_y I_z, \end{aligned} \tag{261}$$

where

$$I_x = \int x_{\mathbf{A}_1}^{l_1} x_{\mathbf{A}_2}^{l_2} x_{\mathbf{A}_3}^{l_3} x_{\mathbf{A}_4}^{l_4} e^{-\gamma_t x_\mathbf{T}^2} dx \tag{262}$$

(I_y and I_z are defined similarly.)

It can be shown that

$$x_{\mathbf{A}_1}^{l_1} x_{\mathbf{A}_2}^{l_2} = \sum_{i=0}^{l_1+l_2} x_{\mathbf{P}}^{i} f_i(l_1, l_2, (\overline{\mathbf{PA}_1})_x, (\overline{\mathbf{PA}_2})_x), \tag{263}$$

where $f_i(l,m,a,b)$ is the coefficient of x^i in the expansion of $(x+a)^l(x+b)^m$. This function is commonly used in the evaluation of integrals over Gaussian primitives in electronic structure calculations. Thus,

$$\begin{aligned} x_{\mathbf{A}_1}^{l_1} x_{\mathbf{A}_2}^{l_2} x_{\mathbf{A}_3}^{l_3} x_{\mathbf{A}_4}^{l_4} &= \sum_{i=0}^{l_1+l_2} x_{\mathbf{P}}^{i} f_i(l_1, l_2, (\overline{\mathbf{PA}_1})_x, (\overline{\mathbf{PA}_2})_x) \\ &\quad \times \sum_{j=0}^{l_3+l_4} x_{\mathbf{Q}}^{j} f_j(l_3, l_4, (\overline{\mathbf{QA}_3})_x, (\overline{\mathbf{QA}_4})_x) \\ &= \sum_{i=0}^{l_1+l_2} f_i(l_1, l_2, (\overline{\mathbf{PA}_1})_x, (\overline{\mathbf{PA}_2})_x) \\ &\quad \times \sum_{j=0}^{l_3+l_4} f_j(l_3, l_4, (\overline{\mathbf{QA}_3})_x, (\overline{\mathbf{QA}_4})_x) \qquad (264) \\ &\quad \times x_{\mathbf{P}}^{i} x_{\mathbf{Q}}^{j} \qquad (265) \end{aligned}$$

Applying Equation 263 to $x_{\mathbf{P}}^{i} x_{\mathbf{Q}}^{j}$, we get

$$\begin{aligned} x_{\mathbf{A}_1}^{l_1} x_{\mathbf{A}_2}^{l_2} x_{\mathbf{A}_3}^{l_3} x_{\mathbf{A}_4}^{l_4} &= \sum_{i=0}^{l_1+l_2} f_i(l_1, l_2, (\overline{\mathbf{PA}_1})_x, (\overline{\mathbf{PA}_2})_x) \\ &\quad \times \sum_{j=0}^{l_3+l_4} f_j(l_3, l_4, (\overline{\mathbf{QA}_3})_x, (\overline{\mathbf{QA}_4})_x) \\ &\quad \times \sum_{k=0}^{i+j} f_k(i, j, (\overline{\mathbf{TP}})_x, (\overline{\mathbf{TQ}})_x) x_{\mathbf{T}}^{k}, \end{aligned} \tag{266}$$

So now, we can further simplify I_x:

$$\begin{aligned} I_x = & \sum_{i=0}^{l_1+l_2} f_i(l_1,l_2,(\overline{\mathbf{PA}_1})_x,(\overline{\mathbf{PA}_2})_x) \\ & \times \sum_{j=0}^{l_3+l_4} f_j(l_3,l_4,(\overline{\mathbf{QA}_3})_x,(\overline{\mathbf{QA}_4})_x) \\ & \times \sum_{k=0}^{i+j} f_k(i,j,(\overline{\mathbf{TP}})_x,(\overline{\mathbf{TQ}})_x) \\ & \times \int x_{\mathbf{T}}^k e^{-\gamma_t x_{\mathbf{T}}^2} dx \end{aligned} \tag{267}$$

Since odd values of k will produce a zero integral, the sum over k can go from zero to $(i+j)/2$, and k can be replaced with $2k$ both places that it occurs. It is also known that

$$\int x_{\mathbf{T}}^{2k} e^{-\gamma_t x_{\mathbf{T}}^2} dx = \frac{(2k-1)!!}{(2\gamma_t)^k} \left(\frac{\pi}{\gamma_t}\right)^{1/2} \tag{268}$$

So now we have all of the necessary ingredients to write down the final form of I_x:

$$\begin{aligned} I_x = & \sum_{i=0}^{l_1+l_2} f_i(l_1,l_2,(\overline{\mathbf{PA}_1})_x,(\overline{\mathbf{PA}_2})_x) \\ & \times \sum_{j=0}^{l_3+l_4} f_j(l_3,l_4,(\overline{\mathbf{QA}_3})_x,(\overline{\mathbf{QA}_3})_x) \\ & \times \sum_{k=0}^{(i+j)/2} f_{2k}(i,j,(\overline{\mathbf{TP}})_x,(\overline{\mathbf{TQ}})_x) \\ & \times \frac{(2k-1)!!}{(2\gamma_t)^k} \left(\frac{\pi}{\gamma_t}\right)^{1/2} \end{aligned} \tag{269}$$

We now have an analytic form for the four-overlap integral that can be coded in an efficient algorithm for computation of NEO-HF and NEO-MP2 annihilation rates. An example of such a code along with the annihilation rate calculation are given the *MChem* program in Appendix B.

A.8 Fourier Transform of the Product of Primitive Gaussian Basis Functions

Given a function f, we define the *Fourier transform pair* as

$$F(p) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i p x} dx, \tag{270}$$

$$f(x) = \int_{-\infty}^{\infty} F(p) e^{2\pi i p x} dp. \tag{271}$$

Equation (270) is called the (direct) Fourier transform and Equation (271) is known as the inverse Fourier transform. In equation form, these are

$$F(p) = \mathcal{F}\left[f(x)\right] \tag{272}$$

$$f(x) = \mathcal{F}^{-1}\left[F(p)\right]. \tag{273}$$

A.8.1 Separable Functions. The wave functions that we will be dealing with are three-dimensional, separable functions. We say a three-dimensional function $f(x,y,z)$ is separable if it can be written as the product of three one-dimensional functions $f_1(x)$, $f_2(y)$, and $f_3(z)$:

$$f(x,y,z) = f_1(x) f_2(y) f_3(z). \tag{274}$$

Theorem 1. *(Separability) If a function f is separable into three functions f_1, f_2, and f_3, and all three possess Fourier transforms given by F_1, F_2, and F_3, respectively, then the three-dimensional Fourier transform of f is simply the product of the three individual transforms:*

$$F(p_x, p_y, p_z) = F_1(p_x) F_2(p_y) F_3(p_z). \tag{275}$$

A.8.2 Other Useful Theorems. The following are some other useful theorems concerning Fourier transforms. [70]

Theorem 2. *(Linearity) If both functions $f(x)$ and $g(x)$ have Fourier transforms $F(p)$ and $G(p)$, respectively, then $h(x) = af(x) + bg(x)$ has Fourier transform $aF(p) + bF(p)$.*

Theorem 3. *(First Shifting Theorem) If $f(x)$ has a Fourier transform $F(p)$, then the Fourier transform of $f(x - x_0)$ is given by $F(p)e^{-2\pi i p x_0}$.*

Thus, a phase shift in the spatial domain gives rise to a sinusoidal-type modulation in the momentum domain. Important to note, however, is that the modulus of the transform is unchanged or invariant:

$$\left|F(p)e^{-2\pi ipx_0}\right| = \left|F(p)\right|\left|e^{-2\pi ipx_0}\right| = \left|F(p)\right| \tag{276}$$

Theorem 4. *(Derivative of the Transform) If $\mathcal{F}[f(x)] = F(p)$ and $\mathcal{F}[x^n f(x)]$ exists, then they are related as follows:*

$$(-2\pi i)^n \mathcal{F}[x^n f(x)] = \frac{d^n F(p)}{dp^n}. \tag{277}$$

From this, we see that the Fourier transform of $x^n f(x)$ is found by taking the n^{th} derivative of $F(p)$ and dividing the result by $(-2\pi i)^n$.

A.8.3 Self-Reciprocity of Gaussian Functions. It is well-known that Gaussian distributions, $f(x) = e^{-\alpha x^2}$, enjoy a special status called "self-reciprocity". This is easily shown [70]:

$$\begin{aligned} F(p) &= \int_{-\infty}^{\infty} e^{-\alpha x^2} e^{-2\pi ipx} dx & (278)\\ &= \int_{-\infty}^{\infty} e^{-\alpha(x^2 + 2\pi ipx/\alpha)} dx & (279) \end{aligned}$$

We can write $x^2 + 2\pi ipx/\alpha$ as $(x + i\pi p/\alpha)^2 + (\pi p/\alpha)^2$.

$$\begin{aligned} F(p) &= \int_{-\infty}^{\infty} e^{-\alpha[(x+i\pi p/\alpha)^2 + (\pi p/\alpha)^2]} dx & (280)\\ &= \int_{-\infty}^{\infty} e^{-\alpha(x+i\pi p/\alpha)^2} e^{-\alpha(\pi p/\alpha)^2} dx & (281)\\ &= e^{-\pi^2 p^2/\alpha} \int_{-\infty}^{\infty} e^{-\alpha(x+i\pi p/\alpha)^2} dx & (282) \end{aligned}$$

Performing a change of variable, $s = x + i\pi p/\alpha$,

$$F(p) = e^{-\pi^2 p^2/\alpha} \int_{-\infty}^{\infty} e^{-\alpha s^2} ds \tag{283}$$

s is complex, and this integral requires contour integration. It can be shown that the integral is $\sqrt{\pi/\alpha}$. Thus, the Fourier transform of the Gaussian distribution is

$$\mathcal{F}[e^{-\alpha x^2}] = \sqrt{\frac{\pi}{\alpha}} e^{-\pi^2 p^2/\alpha}. \tag{284}$$

A.8.4 Fourier Transform of a Molecular Orbital. As shown in equations (80) through (85),

$$\phi_k = \sum_k C_k \varphi_k^{l_k, m_k, n_k}. \tag{285}$$

Since the individual Gaussian primitives are separable, this wavefunction can be rewritten as

$$\begin{aligned} \phi_k = \sum_k & C_k (x - \mathbf{A}_{k,x})^{l_k} e^{-\alpha_k (x - \mathbf{A}_{k,x})^2} \\ & \times (y - \mathbf{A}_{k,y})^{m_k} e^{-\alpha_k (y - \mathbf{A}_{k,y})^2} \\ & \times (z - \mathbf{A}_{k,z})^{n_k} e^{-\alpha_k (z - \mathbf{A}_{k,z})^2} \end{aligned} \tag{286}$$

Now, using Theorems 1, 2, 3, and 4 (in that order), the Fourier transform of the spatial orbital is:

$$\begin{aligned}
\mathcal{F}[\phi_k] =& \sum_k C_k \mathcal{F}\left[(x-\mathbf{A}_{k,x})^{l_k} e^{-\alpha_k (x-\mathbf{A}_{k,x})^2}\right] \\
&\times \mathcal{F}\left[(y-\mathbf{A}_{k,y})^{m_k} e^{-\alpha_k (y-\mathbf{A}_{k,y})^2}\right] \\
&\times \mathcal{F}\left[(z-\mathbf{A}_{k,z})^{n_k} e^{-\alpha_k (z-\mathbf{A}_{k,z})^2}\right] \qquad (287) \\
=& \sum_k C_k e^{-2\pi i p_x \mathbf{A}_{k,x}} \mathcal{F}\left[x^{l_k} e^{-\alpha_k x^2}\right] \\
&\times e^{-2\pi i p_y \mathbf{A}_{k,y}} \mathcal{F}\left[y^{m_k} e^{-\alpha_k y^2}\right] \\
&\times e^{-2\pi i p_z \mathbf{A}_{k,z}} \mathcal{F}\left[z^{n_k} e^{-\alpha_k z^2}\right] \qquad (288) \\
=& \sum_k C_k e^{-2\pi i p_x \mathbf{A}_{k,x}} \frac{1}{(-2\pi i)^{l_k}} \frac{d^{l_k}}{dp_x^{l_k}} \mathcal{F}\left[e^{-\alpha_k x^2}\right] \\
&\times e^{-2\pi i p_y \mathbf{A}_{k,y}} \frac{1}{(-2\pi i)^{m_k}} \frac{d^{m_k}}{dp_y^{m_k}} \mathcal{F}\left[e^{-\alpha_k y^2}\right] \\
&\times e^{-2\pi i p_z \mathbf{A}_{k,z}} \frac{1}{(-2\pi i)^{n_k}} \frac{d^{n_k}}{dp_z^{n_k}} \mathcal{F}\left[e^{-\alpha_k z^2}\right] \qquad (289) \\
=& \sum_k C_k e^{-2\pi i p_x \mathbf{A}_{k,x}} \frac{1}{(-2\pi i)^{l_k}} \sqrt{\frac{\pi}{\alpha_k}} \frac{d^{l_k}}{dp_x^{l_k}} e^{-\pi^2 p_x^2/\alpha_k} \\
&\times e^{-2\pi i p_y \mathbf{A}_{k,y}} \frac{1}{(-2\pi i)^{m_k}} \sqrt{\frac{\pi}{\alpha_k}} \frac{d^{m_k}}{dp_y^{m_k}} e^{-\pi^2 p_y^2/\alpha_k} \\
&\times e^{-2\pi i p_z \mathbf{A}_{k,z}} \frac{1}{(-2\pi i)^{n_k}} \sqrt{\frac{\pi}{\alpha_k}} \frac{d^{n_k}}{dp_z^{n_k}} e^{-\pi^2 p_z^2/\alpha_k} \qquad (290) \\
=& \sum_k C_k e^{-2\pi i p_x \mathbf{A}_{k,x}} \left(\frac{\pi p_x}{i\alpha_k}\right)^{l_k} \sqrt{\frac{\pi}{\alpha_k}} e^{-\pi^2 p_x^2/\alpha_k} \\
&\times e^{-2\pi i p_y \mathbf{A}_{k,y}} \left(\frac{\pi p_y}{i\alpha_k}\right)^{l_k} \sqrt{\frac{\pi}{\alpha_k}} e^{-\pi^2 p_y^2/\alpha_k} \\
&\times e^{-2\pi i p_z \mathbf{A}_{k,z}} \left(\frac{\pi p_z}{i\alpha_k}\right)^{l_k} \sqrt{\frac{\pi}{\alpha_k}} e^{-\pi^2 p_z^2/\alpha_k} \qquad (291)
\end{aligned}$$

A.8.5 Products of Spatial Orbitals. We also need to perform the Fourier transform of the product of two spatial orbitals, and for this, we need to examine the Fourier transform of the product of two Gaussian primitives, $\mathcal{F}\left[\varphi_1^{l_1,m_1,n_1}\varphi_2^{l_2,m_2,n_2}\right]$. Using the Gaus-

sian Product Theorem,

$$\mathcal{F}\left[\varphi_1^{l_1,m_1,n_1}\varphi_2^{l_2,m_2,n_2}\right] = \mathcal{F}\left[x_{\mathbf{A}_1}^{l_1}x_{\mathbf{A}_2}^{l_2}y_{\mathbf{A}_1}^{m_1}y_{\mathbf{A}_2}^{m_2}z_{\mathbf{A}_1}^{n_1}z_{\mathbf{A}_2}^{n_2}\exp\left(-\alpha_1\alpha_2\left(\overline{\mathbf{A}_1\mathbf{A}_2}\right)^2/\gamma_q\right)e^{-\gamma_q\mathbf{r}_\mathbf{Q}^2}\right] \quad (292)$$

where

$$\gamma_q \quad = \alpha_1+\alpha_2, \quad (293)$$

$$\mathbf{Q} \quad = \tfrac{\alpha_1\mathbf{A}_1+\alpha_2\mathbf{A}_2}{\gamma_q}. \quad (294)$$

Separating variables and applying Theorems 1 and 2,

$$\mathcal{F}\left[\varphi_1^{l_1,m_1,n_1}\varphi_2^{l_2,m_2,n_2}\right] = e^{-\alpha_1\alpha_2\left(\overline{\mathbf{A}_1\mathbf{A}_2}\right)^2/\gamma_q}F_xF_yF_z \quad (295)$$

where

$$F_x = \mathcal{F}\left[x_{\mathbf{A}_1}^{l_1}x_{\mathbf{A}_2}^{l_2}e^{-\gamma_q x_\mathbf{Q}^2}\right] \quad (296)$$

Using Equation (263) and applying Theorem 2,

$$F_x = \mathcal{F}\left[\sum_{r=0}^{l_1+l_2}x_\mathbf{Q}^r f_r(l_1,l_2,(\overline{\mathbf{QA}_1})_x,(\overline{\mathbf{QA}_2})_x)e^{-\gamma_q x_\mathbf{Q}^2}\right] \quad (297)$$

$$= \sum_{r=0}^{l_1+l_2} f_r(l_1,l_2,(\overline{\mathbf{QA}_1})_x,(\overline{\mathbf{QA}_2})_x)\mathcal{F}\left[x_\mathbf{Q}^r e^{-\gamma_q x_\mathbf{Q}^2}\right]. \quad (298)$$

Now, rewriting $x_\mathbf{Q}$ as $(x-\mathbf{Q}_x)$ and applying Theorem 3,

$$F_x = \sum_{r=0}^{l_1+l_2} f_r(l_1,l_2,(\overline{\mathbf{QA}_1})_x,(\overline{\mathbf{QA}_2})_x)\mathcal{F}\left[(x-\mathbf{Q}_x)^r e^{-\gamma_q(x-\mathbf{Q}_x)^2}\right] \quad (299)$$

$$= \sum_{r=0}^{l_1+l_2} f_r(l_1,l_2,(\overline{\mathbf{QA}_1})_x,(\overline{\mathbf{QA}_2})_x)e^{-2\pi i p_x\mathbf{Q}_x}\mathcal{F}\left[x^r e^{-\gamma_q x^2}\right] \quad (300)$$

Applying Theorem 4 and Equation (284) (and simplifying),

$$F_x = \sum_{r=0}^{l_1+l_2} f_r(l_1,l_2,(\overline{\mathbf{QA}_1})_x,(\overline{\mathbf{QA}_2})_x) e^{-2\pi i p_x \mathbf{Q}_x} \frac{1}{(-2\pi i)^r} \frac{d^r}{dp_x^r} \mathcal{F}\left[e^{-\gamma_q x^2}\right] \tag{301}$$

$$= \sum_{r=0}^{l_1+l_2} f_r(l_1,l_2,(\overline{\mathbf{QA}_1})_x,(\overline{\mathbf{QA}_2})_x) e^{-2\pi i p_x \mathbf{Q}_x} \frac{1}{(-2\pi i)^r} \frac{d^r}{dp_x^r} \sqrt{\frac{\pi}{\gamma_q}} e^{-\pi^2 p_x^2/\gamma_q} \tag{302}$$

$$= \sum_{r=0}^{l_1+l_2} f_r(l_1,l_2,(\overline{\mathbf{QA}_1})_x,(\overline{\mathbf{QA}_2})_x) e^{-2\pi i p_x \mathbf{Q}_x} \sqrt{\frac{\pi}{\gamma_q}} \frac{1}{(-2\pi i)^r} \left(\frac{-2\pi^2 p_x}{\gamma_q}\right)^r e^{-\pi^2 p_x^2/\gamma_q} \tag{303}$$

$$= \sum_{r=0}^{l_1+l_2} f_r(l_1,l_2,(\overline{\mathbf{QA}_1})_x,(\overline{\mathbf{QA}_2})_x) e^{-2\pi i p_x \mathbf{Q}_x} \sqrt{\frac{\pi}{\gamma_q}} \left(\frac{\pi i p_x}{\gamma_q}\right)^r e^{-\pi^2 p_x^2/\gamma_q} \tag{304}$$

Substituting this result into Equation (295), we get

$$\begin{aligned}\mathcal{F}\left[\varphi_1^{l_1,m_1,n_1}\varphi_2^{l_2,m_2,n_2}\right] &= e^{-\alpha_1\alpha_2\left(\overline{\mathbf{A}_1\mathbf{A}_2}\right)^2/\gamma_q}\sqrt{\frac{\pi}{\gamma_q}} e^{-2\pi i \mathbf{p}\cdot\mathbf{Q}} e^{-\pi^2\mathbf{p}^2/\gamma_q} \\ &\times \sum_{r=0}^{l_1+l_2} f_r(l_1,l_2,(\overline{\mathbf{QA}_1})_x,(\overline{\mathbf{QA}_2})_x)\left(\frac{\pi i p_x}{\gamma_q}\right)^r \\ &\times \sum_{s=0}^{m_1+m_2} f_s(m_1,m_2,(\overline{\mathbf{QA}_1})_y,(\overline{\mathbf{QA}_2})_y)\left(\frac{\pi i p_y}{\gamma_q}\right)^s \\ &\times \sum_{t=0}^{n_1+n_2} f_t(n_1,n_2,(\overline{\mathbf{QA}_1})_z,(\overline{\mathbf{QA}_2})_z)\left(\frac{\pi i p_z}{\gamma_q}\right)^t .\end{aligned} \tag{305}$$

Now, we introduce a notation analogous to that used for the four-overlap integral for the Fourier transform of wavefunction products:

$$G_{ij}^{kl} = \mathcal{F}\left[\phi_i\phi_k\right]^* \mathcal{F}\left[\phi_j\phi_l\right] \tag{306}$$

for spatial orbitals and

$$g_{ij}^{kl} = \mathcal{F}\left[\varphi_i\varphi_k\right]^* \mathcal{F}\left[\varphi_j\varphi_l\right] \tag{307}$$

for Gaussian primitives. In section A.9, it is shown that the two-photon momentum density, $\rho^{2\gamma}(\mathbf{p})$, for a NEO-RHF electron-positron wavefunction is related to the squared modulus of the Fourier transform of the product of two spatial orbitals. This expression can be written

as

$$G_{11}^{22} = \left|\mathcal{F}\left[\phi_1(\mathbf{r})\phi_2(\mathbf{r})\right]\right|^2 \tag{308}$$

$$= \left|\mathcal{F}\left[\sum_j C_j^1 \varphi_j^{l_j,m_j,n_j}(\alpha_j,\mathbf{A}_j)\sum_k C_k^2 \varphi_k^{l_k,m_k,n_k}(\alpha_k,\mathbf{A}_k)\right]\right|^2 \tag{309}$$

$$= \sum_{jk} C_j^1 C_k^2 \left|\mathcal{F}\left[\varphi_j^{l_j,m_j,n_j}(\alpha_j,\mathbf{A}_j)\varphi_k^{l_k,m_k,n_k}(\alpha_k,\mathbf{A}_k)\right]\right|^2 \tag{310}$$

$$= \sum_{jk} C_j^1 C_k^2 g_{jj}^{kk} \tag{311}$$

$$\begin{aligned} = \sum_{jk} C_j^1 C_k^1 \Bigg| & e^{-\alpha_j\alpha_k\left(\overline{\mathbf{A}_j\mathbf{A}_k}\right)^2/\gamma_{jk}}\sqrt{\frac{\pi}{\gamma_{jk}}}\,e^{-2\pi i\mathbf{p}\cdot\mathbf{Q}_{jk}}\,e^{-\pi^2\mathbf{p}^2/\gamma_{jk}} \\ & \times \sum_{r=0}^{l_j+l_k} f_r(l_j,l_k,(\overline{\mathbf{Q}_{jk}\mathbf{A}_j})_x,(\overline{\mathbf{Q}_{jk}\mathbf{A}_k})_x)\left(\frac{\pi i p_x}{\gamma_{jk}}\right)^r \\ & \times \sum_{s=0}^{m_j+m_k} f_s(m_j,m_k,(\overline{\mathbf{Q}_{jk}\mathbf{A}_j})_y,(\overline{\mathbf{Q}_{jk}\mathbf{A}_k})_y)\left(\frac{\pi i p_y}{\gamma_{jk}}\right)^s \\ & \times \sum_{t=0}^{n_j+n_k} f_t(n_j,n_k,(\overline{\mathbf{Q}_{jk}\mathbf{A}_j})_z,(\overline{\mathbf{Q}_{jk}\mathbf{A}_k})_z)\left(\frac{\pi i p_z}{\gamma_{jk}}\right)^t\Bigg|^2, \end{aligned} \tag{312}$$

where

$$\gamma_{jk} = \alpha_j + \alpha_k \tag{313}$$

$$\mathbf{Q}_{jk} = \frac{\alpha_j\mathbf{A}_j+\alpha_k\mathbf{A}_k}{\gamma_{jk}}, \tag{314}$$

and C_j^1 and C_k^2 are the basis function coefficients for orbitals 1 and 2, respectively. Expanding this, we get

$$
\begin{aligned}
G_{11}^{22} = \sum_{jk} \frac{\pi C_j^1 C_k^2}{\gamma_{jk}} \, & e^{-2\alpha_j \alpha_k \left(\overline{\mathbf{A}_j \mathbf{A}_k}\right)^2 / \gamma_{jk}} \, e^{-2\pi^2 \mathbf{p}^2 / \gamma_{jk}} \\
& \times \sum_{r,u=0}^{l_j+l_k} f_r(l_j, l_k, (\overline{\mathbf{Q}_{jk}\mathbf{A}_j})_x, (\overline{\mathbf{Q}_{jk}\mathbf{A}_k})_x) \\
& \times f_u(l_j, l_k, (\overline{\mathbf{Q}_{jk}\mathbf{A}_j})_x, (\overline{\mathbf{Q}_{jk}\mathbf{A}_k})_x) \left(\frac{\pi p_x}{\gamma_{jk}}\right)^{r+u} \\
& \times \sum_{s,v=0}^{m_j+m_k} f_s(m_j, m_k, (\overline{\mathbf{Q}_{jk}\mathbf{A}_j})_y, (\overline{\mathbf{Q}_{jk}\mathbf{A}_k})_y) \\
& \times f_v(m_j, m_k, (\overline{\mathbf{Q}_{jk}\mathbf{A}_j})_y, (\overline{\mathbf{Q}_{jk}\mathbf{A}_k})_y) \left(\frac{\pi p_y}{\gamma_{jk}}\right)^{s+v} \\
& \times \sum_{t,w=0}^{n_j+n_k} f_t(n_j, n_k, (\overline{\mathbf{Q}_{jk}\mathbf{A}_j})_z, (\overline{\mathbf{Q}_{jk}\mathbf{A}_k})_z) \\
& \times f_w(n_j, n_k, (\overline{\mathbf{Q}_{jk}\mathbf{A}_j})_z, (\overline{\mathbf{Q}_{jk}\mathbf{A}_k})_z) \left(\frac{\pi p_z}{\gamma_{jk}}\right)^{t+w} .
\end{aligned}
\tag{315}
$$

A.9 The Two-Photon Momentum Density

From Ref [38], equation (7),

$$\rho^{2\gamma}(\mathbf{p}) = 4\pi r_0^2 cN \int d\mathbf{r}^{\mathrm{e}}_{(1)} \left| \int d\mathbf{r}_1 d\mathbf{r}_{1'} e^{-2\pi i \mathbf{p}\cdot(\mathbf{r}_1+\mathbf{r}_{1'})/2} \delta(\mathbf{r}_{1'}-\mathbf{r}_1) \Psi(\mathbf{r}^{\mathrm{e}}, \mathbf{r}_{1'}) \right|^2 \tag{316}$$

Integrating over the positron coordinates, we get

$$\rho^{2\gamma}(\mathbf{p}) = 4\pi r_0^2 cN \int d\mathbf{r}^{\mathrm{e}}_{(1)} \left| \int d\mathbf{r}_1 e^{-2\pi i \mathbf{p}\cdot\mathbf{r}_1} \Psi(\mathbf{r}^{\mathrm{e}}, \mathbf{r}_1) \right|^2 . \tag{317}$$

Here, $\Psi(\mathbf{r}^{\mathrm{e}}, \mathbf{r}_1)$ is the N-electron, one-positron wavefunction with the coordinates, $\mathbf{r}_1$, substituted for the positron coordinates, $\mathbf{r}_{1'}$.

Substituting the NEO-RHF wavefunction into equation (317),

$$\rho^{2\gamma}(\mathbf{p}) = 4\pi r_0^2 cN \int d\mathbf{r}^{\mathrm{e}}_{(1)} \left| \int d\mathbf{r}_1 e^{-2\pi i \mathbf{p}\cdot\mathbf{r}_1} \Phi^{\mathrm{e}}_0(\mathbf{r}^{\mathrm{e}}) \phi_{1'}(\mathbf{r}_1) \right|^2 \tag{318}$$

$$= 4\pi r_0^2 cN \int d\mathbf{r}^{\mathrm{e}}_{(1)} \left| \int d\mathbf{r}_1 e^{-2\pi i \mathbf{p}\cdot\mathbf{r}_1} \frac{1}{\sqrt{N!}} \sum_i^N (-1)^{i+1} \phi_i(\mathbf{r}_1) \Phi^{\mathrm{e}}_{0\,1i} \phi_{1'}(\mathbf{r}_1) \right|^2 \tag{319}$$

$$= 4\pi r_0^2 cN \int d\mathbf{r}^{\mathrm{e}}_{(1)} \left| \frac{1}{\sqrt{N!}} \sum_i^N (-1)^{i+1} \Phi^{\mathrm{e}}_{0\,1i} \int d\mathbf{r}_1 e^{-2\pi i \mathbf{p}\cdot\mathbf{r}_1} \phi_i(\mathbf{r}_1) \phi_{1'}(\mathbf{r}_1) \right|^2 \tag{320}$$

$$= 4\pi r_0^2 cN \int d\mathbf{r}^{\mathrm{e}}_{(1)} \left| \frac{1}{\sqrt{N!}} \sum_i^N (-1)^{i+1} \Phi^{\mathrm{e}}_{0\,1i} \mathcal{F}[\phi_i(\mathbf{r}_1) \phi_{1'}(\mathbf{r}_1)] \right|^2 \tag{321}$$

$$= \frac{4\pi r_0^2 cN}{N!} \sum_{i,j}^N (-1)^{i+j} \mathcal{F}[\phi_i(\mathbf{r}_1) \phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_j(\mathbf{r}_1) \phi_{1'}(\mathbf{r}_1)] \int d\mathbf{r}^{\mathrm{e}}_{(1)} \Phi^{\mathrm{e}*}_{0\,1i} \Phi^{\mathrm{e}}_{0\,1j} \tag{322}$$

$$= \frac{4\pi r_0^2 c}{(N-1)!} \sum_{i,j}^N (-1)^{i+j} \mathcal{F}[\phi_i(\mathbf{r}_1) \phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_j(\mathbf{r}_1) \phi_{1'}(\mathbf{r}_1)] (N-1)! \delta_{ij} \tag{323}$$

$$= 4\pi r_0^2 c \sum_i^N |\mathcal{F}[\phi_i(\mathbf{r}_1) \phi_{1'}(\mathbf{r}_1)]|^2 \tag{324}$$

$$= 4\pi r_0^2 c \sum_i^N G_{ii}^{1'1'} \tag{325}$$

For NEO-MP2,

$$\Psi(\mathbf{r}^{\mathrm{e}},\mathbf{r}_{1'}) = \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) + \sum_{\substack{a<b\\r<s}} {}^{\mathrm{ee}}C_{ab}^{rs}\Phi_{ab}^{rs}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_{1'}) - \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'}\Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{r'}(\mathbf{r}_{1'}). \quad (326)$$

Substituting the NEO-MP2 first-order wavefunction into equation (317),

$$\begin{aligned}
\rho^{2\gamma}(\mathbf{p}) =& 4\pi r_0^2 cN \int d\mathbf{r}_{(1)}^{\mathrm{e}} \\
&\times \Bigg| \int d\mathbf{r}_1 e^{-2\pi i \mathbf{p}\cdot\mathbf{r}_1} \Phi_0^{\mathrm{e}}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_1) \\
&\quad + \sum_{\substack{a<b\\r<s}} {}^{\mathrm{ee}}C_{ab}^{rs}\Phi_{ab}^{rs}(\mathbf{r}^{\mathrm{e}})\phi_{1'}(\mathbf{r}_1) \\
&\quad - \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'}\Phi_a^r(\mathbf{r}^{\mathrm{e}})\phi_{r'}(\mathbf{r}_1) \Bigg|^2 \qquad (327) \\
=& 4\pi r_0^2 cN \int d\mathbf{r}_{(1)}^{\mathrm{e}} \\
&\times \Bigg| \int d\mathbf{r}_1 e^{-2\pi i \mathbf{p}\cdot\mathbf{r}_1} \frac{1}{\sqrt{N!}} \sum_j^N (-1)^{j+1}\phi_j(\mathbf{r}_1)\Phi_{0\,1j}^{\mathrm{e}}\phi_{1'}(\mathbf{r}_1) \\
&\quad + \sum_{\substack{a<b\\r<s}} {}^{\mathrm{ee}}C_{ab}^{rs}\frac{1}{\sqrt{N!}} \sum_j^N (-1)^{j+1}\psi_{ab\,j}^{rs}(\mathbf{r}_1)\Phi_{ab\,1j}^{rs}\phi_{1'}(\mathbf{r}_1) \\
&\quad - \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'}\frac{1}{\sqrt{N!}} \sum_j^N (-1)^{j+1}\psi_{a\,j}^{r}(\mathbf{r}_1)\Phi_{a\,1j}^{r}\phi_{r'}(\mathbf{r}_1) \Bigg|^2 \qquad (328)
\end{aligned}$$

$$
\begin{aligned}
\rho^{2\gamma}(\mathbf{p}) =& \frac{4\pi r_0^2 c}{(N-1)!}\int d\mathbf{r}^{\mathrm{e}}_{(1)} \\
&\times \left| \sum_j^N (-1)^{j+1} \Phi^{\mathrm{e}}_{0\ 1j} \int d\mathbf{r}_1 e^{-2\pi i \mathbf{p}\cdot\mathbf{r}_1} \phi_j(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1) \right. \\
&\quad + \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C^{rs}_{ab} \sum_j^N (-1)^{j+1} \Phi^{rs}_{ab\ 1j} \int d\mathbf{r}_1 e^{-2\pi i \mathbf{p}\cdot\mathbf{r}_1} \psi^{rs}_{ab\ j}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1) \\
&\quad \left. - \sum_{arr'} {}^{\mathrm{ep}}C^{rr'}_{a1'} \sum_j^N (-1)^{j+1} \Phi^{r}_{a\ 1j} \int d\mathbf{r}_1 e^{-2\pi i \mathbf{p}\cdot\mathbf{r}_1} \psi^{r}_{a\ j}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1) \right|^2
\end{aligned} \tag{329}
$$

$$
\begin{aligned}
=& \frac{4\pi r_0^2 c}{(N-1)!}\int d\mathbf{r}^{\mathrm{e}}_{(1)} \\
&\times \left| \sum_j^N (-1)^{j+1} \left\{ \Phi^{\mathrm{e}}_{0\ 1j} \mathcal{F}[\phi_j(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \right. \right. \\
&\quad + \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C^{rs}_{ab} \Phi^{rs}_{ab\ 1j} \mathcal{F}[\psi^{rs}_{ab\ j}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \\
&\quad \left. \left. - \sum_{arr'} {}^{\mathrm{ep}}C^{rr'}_{a1'} \Phi^{r}_{a\ 1j} \mathcal{F}[\psi^{r}_{a\ j}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)] \right\} \right|^2
\end{aligned} \tag{330}
$$

$$
\begin{aligned}
\rho^{2\gamma}(\mathbf{p}) = & \frac{4\pi r_0^2 c}{(N-1)!} \int d\mathbf{r}^{\mathrm{e}}_{(1)} \sum_{ij}^{N} (-1)^{i+j} \Bigg(\Phi^{\mathrm{e}*}_{\substack{0 \\ 1i}} \Phi^{\mathrm{e}}_{\substack{0 \\ 1j}} \mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_j(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \\
& + 2 \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C^{rs}_{ab} \Phi^{\mathrm{e}*}_{\substack{0 \\ 1i}} \Phi^{rs}_{\substack{ab \\ 1j}} \mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\psi^{rs}_{\substack{ab \\ j}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \\
& - 2 \sum_{arr'} {}^{\mathrm{ep}}C^{rr'}_{a1'} \Phi^{\mathrm{e}*}_{\substack{0 \\ 1i}} \Phi^{r}_{\substack{a \\ 1j}} \mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\psi^{r}_{\substack{a \\ j}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)] \\
& + \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C^{rs}_{ab} \sum_{\substack{c<d \\ t<u}} {}^{\mathrm{ee}}C^{tu}_{cd} \Phi^{rs*}_{\substack{ab \\ 1i}} \Phi^{tu}_{\substack{cd \\ 1j}} \mathcal{F}[\psi^{rs}_{\substack{ab \\ i}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\psi^{tu}_{\substack{cd \\ j}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \\
& - 2 \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C^{rs}_{ab} \sum_{ctt'} {}^{\mathrm{ep}}C^{tt'}_{c1'} \Phi^{rs*}_{\substack{ab \\ 1i}} \Phi^{t}_{\substack{c \\ 1j}} \mathcal{F}[\psi^{rs}_{\substack{ab \\ i}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\psi^{t}_{\substack{c \\ j}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)] \\
& + \sum_{arr'} {}^{\mathrm{ep}}C^{rr'}_{a1'} \sum_{bss'} {}^{\mathrm{ep}}C^{ss'}_{b1'} \Phi^{r*}_{\substack{a \\ 1i}} \Phi^{s}_{\substack{b \\ 1j}} \mathcal{F}[\psi^{r}_{\substack{a \\ i}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]^* \mathcal{F}[\psi^{s}_{\substack{b \\ j}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)] \Bigg)
\end{aligned}
\tag{331}
$$

$$
\begin{aligned}
= & \frac{4\pi r_0^2 c}{(N-1)!} \sum_{ij}^{N} (-1)^{i+j} \Bigg(\mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_j(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \int d\mathbf{r}^{\mathrm{e}}_{(1)} \Phi^{\mathrm{e}*}_{\substack{0 \\ 1i}} \Phi^{\mathrm{e}}_{\substack{0 \\ 1j}} \\
& + 2 \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C^{rs}_{ab} \mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\psi^{rs}_{\substack{ab \\ j}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \cancelto{0}{\int d\mathbf{r}^{\mathrm{e}}_{(1)} \Phi^{\mathrm{e}*}_{\substack{0 \\ 1i}} \Phi^{rs}_{\substack{ab \\ 1j}}} \\
& - 2 \sum_{arr'} {}^{\mathrm{ep}}C^{rr'}_{a1'} \mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\psi^{r}_{\substack{a \\ j}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)] \int d\mathbf{r}^{\mathrm{e}}_{(1)} \Phi^{\mathrm{e}*}_{\substack{0 \\ 1i}} \Phi^{r}_{\substack{a \\ 1j}} \\
& + \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C^{rs}_{ab} \sum_{\substack{c<d \\ t<u}} {}^{\mathrm{ee}}C^{tu}_{cd} \mathcal{F}[\psi^{rs}_{\substack{ab \\ i}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\psi^{tu}_{\substack{cd \\ j}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \int d\mathbf{r}^{\mathrm{e}}_{(1)} \Phi^{rs*}_{\substack{ab \\ 1i}} \Phi^{tu}_{\substack{cd \\ 1j}} \\
& - 2 \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C^{rs}_{ab} \sum_{ctt'} {}^{\mathrm{ep}}C^{tt'}_{c1'} \mathcal{F}[\psi^{rs}_{\substack{ab \\ i}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\psi^{t}_{\substack{c \\ j}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)] \cancelto{0}{\int d\mathbf{r}^{\mathrm{e}}_{(1)} \Phi^{rs*}_{\substack{ab \\ 1i}} \Phi^{t}_{\substack{c \\ 1j}}} \\
& + \sum_{arr'} {}^{\mathrm{ep}}C^{rr'}_{a1'} \sum_{bss'} {}^{\mathrm{ep}}C^{ss'}_{b1'} \mathcal{F}[\psi^{r}_{\substack{a \\ i}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]^* \mathcal{F}[\psi^{s}_{\substack{b \\ j}}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)] \int d\mathbf{r}^{\mathrm{e}}_{(1)} \Phi^{r*}_{\substack{a \\ 1i}} \Phi^{s}_{\substack{b \\ 1j}} \Bigg)
\end{aligned}
\tag{332}
$$

$$
\begin{aligned}
\rho^{2\gamma}(\mathbf{p}) =& \frac{4\pi r_0^2 c}{(N-1)!}\sum_{ij}^{N}(-1)^{i+j}\Bigg(\mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^*\mathcal{F}[\phi_j(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)](N-1)!\delta_{ij} \\
&-2\sum_{arr'}{}^{\mathrm{ep}}C_{a1'}^{rr'}\mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^*\mathcal{F}[\underset{j}{\psi_a^r}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)](N-1)!\delta_{ij}\delta_{ia} \\
&+\sum_{\substack{a<b\\r<s}}{}^{\mathrm{ee}}C_{ab}^{rs}\sum_{\substack{c<d\\t<u}}{}^{\mathrm{ee}}C_{cd}^{tu}\mathcal{F}[\underset{i}{\psi_{ab}^{rs}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^*\mathcal{F}[\underset{j}{\psi_{cd}^{tu}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \\
&\times(N-1)!(\delta_{ij}\delta_{ac}\delta_{bd}\delta_{rt}\delta_{su}+\delta_{ij}\delta_{ia}\delta_{jc}\delta_{bd}\delta_{su}+\delta_{ij}\delta_{ib}\delta_{jd}\delta_{ac}\delta_{rt}) \\
&+\sum_{arr'}{}^{\mathrm{ep}}C_{a1'}^{rr'}\sum_{bss'}{}^{\mathrm{ep}}C_{b1'}^{ss'}\mathcal{F}[\underset{i}{\psi_a^r}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]^*\mathcal{F}[\underset{j}{\psi_b^s}(\mathbf{r}_1)\phi_{s'}(\mathbf{r}_1)] \\
&\times(N-1)!(\delta_{ij}\delta_{ab}\delta_{rs}+\delta_{ij}\delta_{ia}\delta_{jb})\Bigg)
\end{aligned}
\tag{333}
$$

$$
\begin{aligned}
=&4\pi r_0^2 c\sum_{i}^{N}\Bigg(|\mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]|^2 \\
&-2\sum_{arr'}{}^{\mathrm{ep}}C_{a1'}^{rr'}\mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^*\mathcal{F}[\underset{i}{\psi_a^r}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]\delta_{ia} \\
&+\sum_{\substack{a<b\\r<s}}{}^{\mathrm{ee}}C_{ab}^{rs}\sum_{\substack{c<d\\t<u}}{}^{\mathrm{ee}}C_{cd}^{tu}\mathcal{F}[\underset{i}{\psi_{ab}^{rs}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^*\mathcal{F}[\underset{i}{\psi_{cd}^{tu}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \\
&\times(\delta_{ac}\delta_{bd}\delta_{rt}\delta_{su}+\delta_{ia}\delta_{ic}\delta_{bd}\delta_{su}+\delta_{ib}\delta_{id}\delta_{ac}\delta_{rt}) \\
&+\sum_{arr'}{}^{\mathrm{ep}}C_{a1'}^{rr'}\sum_{bss'}{}^{\mathrm{ep}}C_{b1'}^{ss'}\mathcal{F}[\underset{i}{\psi_a^r}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]^*\mathcal{F}[\underset{i}{\psi_b^s}(\mathbf{r}_1)\phi_{s'}(\mathbf{r}_1)]\,(\delta_{ab}\delta_{rs}+\delta_{ia}\delta_{ib})\Bigg)
\end{aligned}
\tag{334}
$$

$$
\begin{aligned}
\rho^{2\gamma}(\mathbf{p}) =& 4\pi r_0^2 c \Bigg(\sum_i^N |\mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]|^2 \\
& - 2 \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \mathcal{F}[\phi_a(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_r(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)] \\
& + \sum_i^N \sum_{\substack{a<b \\ r<s}} ({}^{\mathrm{ee}}C_{ab}^{rs})^2 \left| \mathcal{F}[\underset{i}{\psi_{ab}^{rs}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \right|^2 \\
& + \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C_{ab}^{rs} \sum_{\substack{c<d \\ t<u}} {}^{\mathrm{ee}}C_{cd}^{tu} \mathcal{F}[\phi_r(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\underset{a}{\psi_{cd}^{tu}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)\delta_{ac}\delta_{bd}\delta_{su} \\
& + \sum_{\substack{a<b \\ r<s}} {}^{\mathrm{ee}}C_{ab}^{rs} \sum_{\substack{c<d \\ t<u}} {}^{\mathrm{ee}}C_{cd}^{tu} \mathcal{F}[\phi_s(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\underset{b}{\psi_{cd}^{tu}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]\delta_{bd}\delta_{ac}\delta_{rt} \\
& + \sum_i^N \sum_{arr's'} {}^{\mathrm{ep}}C_{a1'}^{rr'} {}^{\mathrm{ep}}C_{a1'}^{rs'} \mathcal{F}[\underset{i}{\psi_a^r}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]^* \mathcal{F}[\underset{i}{\psi_a^r}(\mathbf{r}_1)\phi_{s'}(\mathbf{r}_1)] \\
& + \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \sum_{bss'} {}^{\mathrm{ep}}C_{b1'}^{ss'} \mathcal{F}[\phi_r(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]^* \mathcal{F}[\underset{a}{\psi_b^s}(\mathbf{r}_1)\phi_{s'}(\mathbf{r}_1)]\delta_{ab} \Bigg)
\end{aligned} \tag{335}
$$

$$
\begin{aligned}
=& 4\pi r_0^2 c \Bigg(\sum_i^N |\mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]|^2 \\
& - 2 \sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \mathcal{F}[\phi_a(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_r(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)] \\
& + \sum_i^N \sum_{\substack{a<b \\ r<s}} ({}^{\mathrm{ee}}C_{ab}^{rs})^2 \left| \mathcal{F}[\underset{i}{\psi_{ab}^{rs}}(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \right|^2 \\
& + \sum_{\substack{a<b \\ r<s \\ t<u}} {}^{\mathrm{ee}}C_{ab}^{rs} {}^{\mathrm{ee}}C_{ab}^{tu} \mathcal{F}[\phi_r(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_t(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)\delta_{su} \\
& + \sum_{\substack{a<b \\ r<s \\ t<u}} {}^{\mathrm{ee}}C_{ab}^{rs} {}^{\mathrm{ee}}C_{ab}^{tu} \mathcal{F}[\phi_s(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_u(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]\delta_{rt} \\
& + \sum_i^N \sum_{arr's'} {}^{\mathrm{ep}}C_{a1'}^{rr'} {}^{\mathrm{ep}}C_{a1'}^{rs'} \mathcal{F}[\underset{i}{\psi_a^r}(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]^* \mathcal{F}[\underset{i}{\psi_a^r}(\mathbf{r}_1)\phi_{s'}(\mathbf{r}_1)] \\
& + \sum_{arsr's'} {}^{\mathrm{ep}}C_{a1'}^{rr'} {}^{\mathrm{ep}}C_{a1'}^{ss'} \mathcal{F}[\phi_r(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_s(\mathbf{r}_1)\phi_{s'}(\mathbf{r}_1)] \Bigg)
\end{aligned} \tag{336}
$$

$$
\begin{aligned}
\rho^{2\gamma}(\mathbf{p}) =& 4\pi r_0^2 c \Bigg(\sum_i^N |\mathcal{F}[\phi_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]|^2 \\
&- 2\sum_{arr'} {}^{\mathrm{ep}}C_{a1'}^{rr'} \mathcal{F}[\phi_a(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_r(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)] \\
&+ \sum_i^N \sum_{\substack{a<b \\ r<s}} ({}^{\mathrm{ee}}C_{ab}^{rs})^2 \left| \mathcal{F}[\psi_{ab}^{rs}_i(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \right|^2 \\
&+ \sum_{\substack{a<b \\ r,t<s}} {}^{\mathrm{ee}}C_{ab}^{rs}\, {}^{\mathrm{ee}}C_{ab}^{ts} \mathcal{F}[\phi_r(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_t(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1) \\
&+ \sum_{\substack{a<b \\ r<s,u}} {}^{\mathrm{ee}}C_{ab}^{rs}\, {}^{\mathrm{ee}}C_{ab}^{ru} \mathcal{F}[\phi_s(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_u(\mathbf{r}_1)\phi_{1'}(\mathbf{r}_1)] \\
&+ \sum_i^N \sum_{arr's'} {}^{\mathrm{ep}}C_{a1'}^{rr'}\, {}^{\mathrm{ep}}C_{a1'}^{rs'} \mathcal{F}[\psi_{a}^{r}_i(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]^* \mathcal{F}[\psi_{a}^{r}_i(\mathbf{r}_1)\phi_{s'}(\mathbf{r}_1)] \\
&+ \sum_{arsr's'} {}^{\mathrm{ep}}C_{a1'}^{rr'}\, {}^{\mathrm{ep}}C_{a1'}^{ss'} \mathcal{F}[\phi_r(\mathbf{r}_1)\phi_{r'}(\mathbf{r}_1)]^* \mathcal{F}[\phi_s(\mathbf{r}_1)\phi_{s'}(\mathbf{r}_1)] \Bigg)
\end{aligned}
\tag{337}
$$

Appendix B. MChem

1. The *MChem* program is a collection of functions for doing quantum chemistry calculations in *Mathematica* using *MathLink*. It is written in the `CWEB` [29] literate programming language, and the `CWEB` class file is utilized for making the pretty output for inclusion in LaTeX. Many of the functions were adapted from the *PyQuante* program. [41] In addition to the excellent source code documentation provided within the *PyQuante* program, Szabo and Ostlund [63] was referenced for basic theory, Taketa *et al.* [66] for analytic solutions to many of the integrals, and Cook [10] for implementation, especially integral storage.

Although much slower than the fortran code written for inclusion in `GAMESS`, this code is much easier to understand and debug. To save space, only the functions for calculating the four-overlap integrals and the HF annihilation rate are included. The complete *MChem* package includes functions and routines for computing the MP2 energies and annihilation rates, including the electron-electron and electron-positron terms individually.

2. The structure of MChem:

⟨ Header files 3 ⟩
⟨ Global variables 4 ⟩
⟨ Main program 5 ⟩
⟨ Integral functions 6 ⟩
⟨ Auxiliary functions 8 ⟩
⟨ Matrix functions 13 ⟩
⟨ MathLink information 29 ⟩
⟨ MathLink Input/Output 15 ⟩

3. Header files. *MChem* requires a few standard header files from C++ as well as the *mathlink.h* header file that is included with Mathematica.

⟨ Header files 3 ⟩ ≡
```
#include <assert.h>
#include <mathlink.h>
#include <math.h>
#include <float.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

This code is used in section 2.

4. Global variables. Some global variables are defined. `MAX_BAS` and `MAX_MO` can be redefined as necessary.

⟨ Global variables 4 ⟩ ≡
```
#define ZERO  1.0·10^-10      /*used in some tests to avoid divide-by-zero errors */
#define PI  3.1415926535897932385      /*the value of π */
```

```
#define MAX_BAS  128    /* maximum number of basis functions that can be used */
#define MAX_MO   128    /* maximum number of molecular orbitals that can be used */
```

This code is used in section 2.

5. Main program. The main program is standardized for all MathLink programs.

```
⟨ Main program 5 ⟩ ≡
  int main(int argc, char *argv[])
  {
    argc = argc;
    return MLMain(0, argv);
  }
```

This code is used in section 2.

6. Integral functions. The complete *MChem* program includes all of the integrals necessary to compute the Hartree-Fock and MP2 energies and annihilation rates. Here, only the four-overlap integral is included.

⟨ Integral functions 6 ⟩ ≡
⟨ FourOverlap 7 ⟩

This code is used in section 2.

7. The *FourOverlap* function computes the four-overlap integral that is used in the calculation of the electron-positron annihilation rate. It is the integral over two electronic and two positronic molecular orbitals:

$$s_{ijj'j'} = \int \phi_i(\mathbf{r})\phi_j(\mathbf{r})\phi_{i'}(\mathbf{r})\phi_{j'}(\mathbf{r})d\mathbf{r} \tag{338}$$

⟨ FourOverlap 7 ⟩ ≡

```
double FourOverlap(double ExpI, double ExpJ, double ExpK, double ExpL, double
        NormI, double NormJ, double NormK, double NormL, int lI[3], long
        lIlen, int lJ[3], long lJlen, int lK[3], long lKlen, int lL[3], long lLlen, double
        rI[3], long rIlen, double rJ[3], long rJlen, double rK[3], long rKlen, double
        rL[3], long rLlen)
{
  int i, j, k, m;
  double gammap, gammaq, gammat;
  double P[3], Q[3], T[3], PrI[3], PrJ[3], QrK[3], QrL[3], TP[3], TQ[3];
  double rIJ[3], rKL[3], PQ[3];
  double rIrJsq = 0., rKrLsq = 0., PQsq = 0.;
  double I[3], iI, jI;
  double x1, x2, x3, x4;
  gammap = ExpI + ExpJ;
  gammaq = ExpK + ExpL;
  gammat = gammap + gammaq;
  for (k = 0; k < 3; k++) {
    P[k] = (ExpI * rI[k] + ExpJ * rJ[k])/gammap;
    Q[k] = (ExpK * rK[k] + ExpL * rL[k])/gammaq;
    T[k] = (gammap * P[k] + gammaq * Q[k])/gammat;
    rIJ[k] = rI[k] - rJ[k];
    rKL[k] = rK[k] - rL[k];
    PQ[k] = P[k] - Q[k];
    PrI[k] = P[k] - rI[k];
    PrJ[k] = P[k] - rJ[k];
    QrK[k] = Q[k] - rK[k];
    QrL[k] = Q[k] - rL[k];
    TP[k] = T[k] - P[k];
```

```
    TQ[k] = T[k] − Q[k];
    I[k] = 0.;
  }
  rIrJsq = rIJ[0] ∗ rIJ[0] + rIJ[1] ∗ rIJ[1] + rIJ[2] ∗ rIJ[2];
  rKrLsq = rKL[0] ∗ rKL[0] + rKL[1] ∗ rKL[1] + rKL[2] ∗ rKL[2];
  PQsq = PQ[0] ∗ PQ[0] + PQ[1] ∗ PQ[1] + PQ[2] ∗ PQ[2];
  for (m = 0; m < 3; m++) {
    for (i = 0; i ≤ lI[m] + lJ[m]; i++) {
      iI = prefactor(i, lI[m], lJ[m], PrI[m], PrJ[m]);
      for (j = 0; j ≤ lK[m] + lL[m]; j++) {
        jI = prefactor(j, lK[m], lL[m], QrK[m], QrL[m]);
        for (k = 0; k ≤ (i + j)/2; k++) {
          I[m] += iI ∗ jI ∗ prefactor(2 ∗ k, i, j, TP[m],
              TQ[m]) ∗ sqrt(PI/gammat) ∗ factorial2(2 ∗ k − 1)/pow(2. ∗ gammat, k);
        }
      }
    }
  }
  x1 = NormI ∗ NormJ ∗ NormK ∗ NormL ∗ I[0] ∗ I[1] ∗ I[2];
  x2 = exp(−1.0 ∗ ExpI ∗ ExpJ ∗ rIrJsq/gammap);
  x3 = exp(−1.0 ∗ ExpK ∗ ExpL ∗ rKrLsq/gammaq);
  x4 = exp(−1.0 ∗ gammap ∗ gammaq ∗ PQsq/gammat);
  return x1 ∗ x2 ∗ x3 ∗ x4;
}
```

This code is used in section 6.

8. Auxiliary functions. There are several *Auxiliaryfunctions* used in quantum chemistry codes, many of which are part of *MChem*. A few of those functions are required by the *FourOverlap* function and are listed here.

⟨ Auxiliary functions 8 ⟩ ≡
  ⟨ prefactor 9 ⟩
  ⟨ factorial1 10 ⟩
  ⟨ factorial2 11 ⟩
  ⟨ binomial 12 ⟩

This code is used in section 2.

9. The function *prefactor* computes the coefficient of x^j in $(x-a)^p(x-b)^q$.

⟨ prefactor 9 ⟩ ≡
```
double prefactor(int j, int p, int q, double a, double b)
{
  int r, rmax;
  double pow1 = 0., pow2 = 0., frslt = 0.;
  rmax = (j < p) ? j : p;
  if (j > p+q) frslt = 0.;
  else {
    for (r = 0; r ≤ rmax; r++) {
      pow1 = (fabs(a) < ZERO ∧ (p−r) < 0) ? 0. : pow(a,(p−r));
      pow2 = (fabs(b) < ZERO ∧ (q+r−j) < 0) ? 0. : pow(b,(q+r−j));
      frslt += pow1 * pow2 * binomial(p,r) * binomial(q, j−r);
    }
  }
  return frslt;
}
```

This code is used in section 8.

10. The function *factorial1* computes $n!$.

⟨ factorial1 10 ⟩ ≡
```
int factorial1(int n)
{
  return (n ≤ 1) ? 1 : (n * factorial1(n−1));
}
```

This code is used in section 8.

11. The function *factorial2* computes $n!!$.

⟨ factorial2 11 ⟩ ≡
```
int factorial2(int n)
{
```

```
  return (n ≤ 1) ? 1 : (n * factorial2(n − 2));
}
```

This code is used in section 8.

12. The function *binomial* computes the binomial coefficient,

$$\binom{a}{b} = \frac{a!}{(a-b)!b!} \tag{339}$$

⟨ binomial 12 ⟩ ≡

```
int binomial(int a, int b)
{
  return (a ≥ b) ? factorial1(a)/(factorial1(a − b) * factorial1(b)) : 0;
}
```

This code is used in section 8.

13. Matrix functions. Several routines in *MChem* are organized under the umbrella *MatrixFunctions* because they all perform loops over integrals. For instance, the *LambdaHF* routine, which is included here, computes the HF electron-positron annihilation rate by looping over the four-overlap integrals calculated in the *FourOverlap* function.

⟨ Matrix functions 13 ⟩ ≡
  ⟨ LambdaHF 14 ⟩

This code is used in section 2.

14. *LambdaHF* computes the Hartree-Fock annihilation rate:

$$\lambda^{\mathrm{HF}} = 2\pi r_0^2 c \sum_{i=1}^{N} S_{ii1'1'} \tag{340}$$

where $S_{ii1'1'}$ is the four-overlap integral involving the ith doubly-occupied electronic molecular orbital (MO) and the occupied positron MO, r_0 is the classical electron radius, and c is the speed of light.

```
⟨ LambdaHF 14 ⟩ ≡
  void LambdaHF(void)
  {
    int i, m, n, o, p;
    double mnopsum = 0.;
    double lambdahf = 0.;
    ⟨ ML Get ebasis 16 ⟩;
    ⟨ ML Get pbasis 17 ⟩;
    ⟨ ML Get nocc 26 ⟩;
    ⟨ ML Get eExp 18 ⟩;
    ⟨ ML Get pExp 19 ⟩;
    ⟨ ML Get eNorm 20 ⟩;
    ⟨ ML Get pNorm 21 ⟩;
    ⟨ ML Get eA 22 ⟩;
    ⟨ ML Get pA 23 ⟩;
    ⟨ ML Get eL 24 ⟩;
    ⟨ ML Get pL 25 ⟩;
    ⟨ ML Get evec 27 ⟩;
    ⟨ ML Get pvec 28 ⟩;
    for (i = 0; i < nocc; i++) {
      for (m = 0; m < ebasis; m++) {
        if (fabs(evec[i][m]) > ZERO) {
          for (n = 0; n < ebasis; n++) {
            if (fabs(evec[i][n]) > ZERO) {
              for (o = 0; o < pbasis; o++) {
                if (fabs(pvec[0][o]) > ZERO) {
                  for (p = 0; p < pbasis; p++) {
```

```
if (fabs(pvec[0][p]) > ZERO) {
    lambdahf += evec[i][m] * evec[i][n] * pvec[0][o] * pvec[0][p] *
        FourOverlap(eExp[m], eExp[n], pExp[o], pExp[p],
        eNorm[m], eNorm[n], pNorm[o], pNorm[p], eL[m], 3, eL[n],
        3, pL[o], 3, pL[p], 3, eA[m], 3, eA[n], 3, pA[o], 3, pA[p], 3);
}
}
}
}
}
}
}
}
}    /* lambdahf=5.0469693435422D+10*lambdahf; */
lambdahf = 1.0093938687084 · 10^+11 * lambdahf;
MLPutDouble(stdlink, lambdahf);
return;
}
```

This code is used in section 13.

15. MathLink Input/Output. The following are MathLink input and output code snippets that are used repeatedly in *MChem*.

⟨ MathLink Input/Output 15 ⟩ ≡
```
⟨ ML Get ebasis 16 ⟩
⟨ ML Get pbasis 17 ⟩
⟨ ML Get eExp 18 ⟩
⟨ ML Get pExp 19 ⟩
⟨ ML Get eNorm 20 ⟩
⟨ ML Get pNorm 21 ⟩
⟨ ML Get eA 22 ⟩
⟨ ML Get pA 23 ⟩
⟨ ML Get eL 24 ⟩
⟨ ML Get pL 25 ⟩
⟨ ML Get nocc 26 ⟩
⟨ ML Get evec 27 ⟩
⟨ ML Get pvec 28 ⟩
```

This code is used in section 2.

16. ML Get ebasis

⟨ ML Get ebasis 16 ⟩ ≡
```
long ebasis;
MLGetLongInteger(stdlink, &ebasis);
```

This code is used in sections 14 and 15.

17. ML Get pbasis

⟨ ML Get pbasis 17 ⟩ ≡
```
long pbasis;
MLGetLongInteger(stdlink, &pbasis);
```

This code is used in sections 14 and 15.

18. ML Get eExp

⟨ ML Get eExp 18 ⟩ ≡
```
int i_eExp;
double eExp[MAX_BAS];
long eExplen;
MLCheckFunction(stdlink, "List", &eExplen);
for (i_eExp = 0; i_eExp < eExplen; i_eExp++) {
  MLGetReal(stdlink, &eExp[i_eExp]);
}
```

This code is used in sections 14 and 15.

19. ML Get pExp

⟨ ML Get pExp 19 ⟩ ≡

```
int i_pExp;
double pExp[MAX_BAS];
long pExplen;
MLCheckFunction(stdlink,"List",&pExplen);
for (i_pExp = 0; i_pExp < pExplen; i_pExp++) {
  MLGetReal(stdlink,&pExp[i_pExp]);
}
```

This code is used in sections 14 and 15.

20. ML Get eNorm

⟨ ML Get eNorm 20 ⟩ ≡

```
int i_eNorm;
long eNormlen;
double eNorm[MAX_BAS];
MLCheckFunction(stdlink,"List",&eNormlen);
for (i_eNorm = 0; i_eNorm < eNormlen; i_eNorm++) {
  MLGetReal(stdlink,&eNorm[i_eNorm]);
}
```

This code is used in sections 14 and 15.

21. ML Get pNorm

⟨ ML Get pNorm 21 ⟩ ≡

```
int i_pNorm;
long pNormlen;
double pNorm[MAX_BAS];
MLCheckFunction(stdlink,"List",&pNormlen);
for (i_pNorm = 0; i_pNorm < pNormlen; i_pNorm++) {
  MLGetReal(stdlink,&pNorm[i_pNorm]);
}
```

This code is used in sections 14 and 15.

22. ML Get eA

⟨ ML Get eA 22 ⟩ ≡

```
int i_eA, j_eA;
long eAlena, eAlenb;
double eA[MAX_BAS][3];
MLCheckFunction(stdlink,"List",&eAlena);
for (i_eA = 0; i_eA < eAlena; i_eA++) {
  MLCheckFunction(stdlink,"List",&eAlenb);
```

```
    for (j_eA = 0; j_eA < eAlenb; j_eA++) {
      MLGetReal(stdlink, &eA[i_eA][j_eA]);
    }
  }
```

This code is used in sections 14 and 15.

23. ML Get pA

```
⟨ ML Get pA 23 ⟩ ≡
  int i_pA, j_pA;
  long pAlena, pAlenb;
  double pA[MAX_BAS][3];
  MLCheckFunction(stdlink, "List", &pAlena);
  for (i_pA = 0; i_pA < pAlena; i_pA++) {
    MLCheckFunction(stdlink, "List", &pAlenb);
    for (j_pA = 0; j_pA < pAlenb; j_pA++) {
      MLGetReal(stdlink, &pA[i_pA][j_pA]);
    }
  }
```

This code is used in sections 14 and 15.

24. ML Get eL

```
⟨ ML Get eL 24 ⟩ ≡
  int i_eL, j_eL;
  long eLlena, eLlenb;
  int eL[MAX_BAS][3];
  MLCheckFunction(stdlink, "List", &eLlena);
  for (i_eL = 0; i_eL < eLlena; i_eL++) {
    MLCheckFunction(stdlink, "List", &eLlenb);
    for (j_eL = 0; j_eL < eLlenb; j_eL++) {
      MLGetInteger(stdlink, &eL[i_eL][j_eL]);
    }
  }
```

This code is used in sections 14 and 15.

25. ML Get pL

```
⟨ ML Get pL 25 ⟩ ≡
  int i_pL, j_pL;
  long pLlena, pLlenb;
  int pL[MAX_BAS][3];
```

```
MLCheckFunction(stdlink,"List",&pLlena);
for (i_pL = 0; i_pL < pLlena; i_pL++) {
  MLCheckFunction(stdlink,"List",&pLlenb);
  for (j_pL = 0; j_pL < pLlenb; j_pL++) {
    MLGetInteger(stdlink,&pL[i_pL][j_pL]);
  }
}
```

This code is used in sections 14 and 15.

26. ML Get nocc

```
⟨ML Get nocc 26⟩ ≡
  long nocc;
  MLGetLongInteger(stdlink,&nocc);
```

This code is used in sections 14 and 15.

27. ML Get evec

```
⟨ML Get evec 27⟩ ≡
  int i_evec, j_evec, eveclena, eveclenb;
  double evec[MAX_MO][MAX_BAS];
  MLCheckFunction(stdlink,"List",&eveclena);
  for (i_evec = 0; i_evec < eveclena; i_evec++) {
    MLCheckFunction(stdlink,"List",&eveclenb);
    for (j_evec = 0; j_evec < eveclenb; j_evec++) {
      MLGetReal(stdlink,&evec[i_evec][j_evec]);
    }
  }
```

This code is used in sections 14 and 15.

28. ML Get pvec

```
⟨ML Get pvec 28⟩ ≡
  int i_pvec, j_pvec, pveclena, pveclenb;
  double pvec[MAX_MO][MAX_BAS];
  MLCheckFunction(stdlink,"List",&pveclena);
  for (i_pvec = 0; i_pvec < pveclena; i_pvec++) {
    MLCheckFunction(stdlink,"List",&pveclenb);
    for (j_pvec = 0; j_pvec < pveclenb; j_pvec++) {
      MLGetReal(stdlink,&pvec[i_pvec][j_pvec]);
    }
  }
```

This code is used in sections 14 and 15.

29. MathLink Information. The link between *Mathematica* and *MChem* is provided by the code in this section.

⟨ MathLink information 29 ⟩ ≡
  ⟨ ML FourOverlap 30 ⟩
  ⟨ ML LambdaHF 31 ⟩

This code is used in section 2.

30. ML FourOverlap.

⟨ ML FourOverlap 30 ⟩ ≡
```
:Begin:
:Function:      FourOverlap
:Pattern:       FourOverlap[
                ExpA_Real,ExpB_Real,ExpG_Real,ExpH_Real,
                NormA_Real,NormB_Real,NormG_Real,NormH_Real,
                lA_List,lB_List,lG_List,lH_List,
                rA_List,rB_List,rG_List,rH_List]
:Arguments:     {ExpA,ExpB,ExpG,ExpH,
                NormA,NormB,NormG,NormH,lA,lB,lG,lH,rA,rB,rG,rH}
:ArgumentTypes: {Real,Real,Real,Real,
                Real,Real,Real,Real,
                IntegerList,IntegerList,IntegerList,IntegerList,
                RealList,RealList,RealList,RealList}
:ReturnType:    Real
:End:
```

This code is used in section 29.

31. ML LambdaHF.

⟨ ML LambdaHF 31 ⟩ ≡
```
:Begin:
:Function:      LambdaHF
:Pattern:       LambdaHF[ebasis_Integer,
                pbasis_Integer,nocc_Integer,eExp_List,pExp_List,
                eNorm_List,pNorm_List,eA_?MatrixQ,pA_?MatrixQ,
                eL_?MatrixQ,pL_?MatrixQ,
                evec_?MatrixQ,pvec_?MatrixQ]
:Arguments:     {ebasis,pbasis,nocc,eExp,pExp,eNorm,pNorm,
                eA,pA,eL,pL,evec,pvec}
:ArgumentTypes: {Manual}
:ReturnType:    Manual
:End:
```

This code is used in section 29.

Appendix C. Modifications of GAMESS and NEO for Positrons

C.1 NEOPOS.SRC

The NEOPOS.SRC module contains all of the completely new routines and functions added to NEO for modeling positrons. It contains the *lambdahf*, *getlambdamp2*, *neodftint*, *neofourc*, and *neoforscr* subroutines. As with the rest of GAMESS, support for 32- and 64-bit systems is incorporated by utilyzing the functionality of the *actvte.f* program, which activates the lines that begin with `*32` and `*64`, as appropriate.

C.1.1 lambdahf subroutine. The main subroutine, *lambdahf*, is given in Listing C.1. It calls the *getlambdamp2* subroutine for calculation of the NEO-HF and NEO-MP2 annihilation rates.

Listing C.1: *lambdahf* subroutine.

```
C*MODULE neopos  *DECK lambdahf
      subroutine lambdahf
c
      implicit double precision(a-h,o-z)
c
      logical pack2e
c
      parameter (maxneo=40)
c
      common /fmcom / x(1)
      common /mp2par/ ospt,tol,method,nwdmp2,mpprop,nacore,
     *                nbcore,noa,nob,no,nbf,nomit,mocphf,maxitc
      common /intfil/ nintmx,nhex,ntupl,pack2e,inttyp,igrdtyp
      common /iofile/ ir,iw,ip,is,ipk,idaf,nav,ioda(950)
      common /nucmoi/ nuniqn,iuniqn(maxneo),iuniqt(maxneo),nqmnuc,
     *                iqmnuc(maxneo),iqntyp(maxneo),numnb,nucst,
     *                nauxnb,iauxnb(maxneo),numult,nna,nnb,ntauxb
      common /neompn/ nemplv
c
c     pire2c is πr_e^2c
      parameter (pire2c=5.0469693435422d+10)
c
      noc=noa
      nvir=no-noc
      nnoc=1
      nnorb=numnb
      nnvir=nnorb-nnoc
c
      write(iw,9000) noc,noa,nacore,nvir,no,nbf,nnoc,nnorb,nnvir,
     *   numnb
      call flshbf(iw)
c
```

```
      call valfm(loadfm)
      loxx=loadfm+1
      loix=loxx+nintmx
      lepxx=loix+nintmx
      lepix=lepxx+nintmx
      lcep=lepix+4*nintmx
      lsep=lcep+noc*no*nnorb
      ivec=lsep+noc*no*nnorb
      ivecn=ivec+no*nbf
      ieng=ivecn+nnorb*numnb
      iengn=ieng+no
      last=iengn+nnorb
      needa=last-loadfm-1
      call getfm(needa)
c
      xlhf=0.0d+00
      xlep=0.0d+00
c
      call daread(idaf,ioda,x(ivec),no*nbf,15,0)
      call daread(idaf,ioda,x(ivecn),nnorb*numnb,444,0)
c
c     dftints contains the four-overlap integrals
c
      is=21
      call seqopn(is,'dftints','unknown',.false.,'unformatted')
c
c     CI annihilation rate not implemented, so we set it to an
c     absurd number
c
      if(nemplv.eq.2) then
         call daread(idaf,ioda,x(ieng),nbf,17,0)
         call daread(idaf,ioda,x(iengn),numnb,447,0)
         it=67
         call seqopn(it,'elnuint','unknown',.false.,'unformatted')
         call getlambdamp2(is,it,xlhf,xlep,x(ivec),x(ivecn),x(iend),
     *      x(iengn),x(lepxx),x(lepix),nintmx,x(lcep),x(lsep))
      else
         xlfci = 9999.9d+99
      end if
c
      call retfm(needa)
c
      xlhf=pire2c*xlhf
      xtauhf=1.0d+00/xlhf
c
      if(nemplv.eq.2) then
c
c        remember that the correction to the annihilation rate is
c        negative therefore the -1 multiplier
c
         xlep=-1.0d+00*pire2c*xlep
         xlmp2=xlhf+xlep
```

```
         xtaump2=1.0d+00/xlmp2
         write(iw,9010) xlhf*1.0d-09,xtauhf*1.0d+09,xlep*1.0d-09,
     *      xlmp2*1.0d-09,xtaump2*1.0d+09
       else
         xlfci=pire2c*xlfci
         xtaufci=1.0d+00/xlfci
         write(iw,9020) xlhf*1.0d-09,xtauhf*1.0d+09,
     *      xlfci*1.0d-09,xtaufci*1.0d+09
       end if
c
       return
 9000 format(/1x,39('-'),3x,23(1h-)/
     *        1x,'neo positron-electron annihilation rate',
     *        3x,'adapted by paul adamson'/
     *        1x,39('-'),3x,23(1h-)/
     *        1x,'-electron-'/
     *        1x,'noc  =',i4,' noa  =',i5,' nacore=',i5,' nvir =',i5/
     *        1x,'no   =',i4,' nbf  =',i5/
     *        1x,'-positron-'/
     *        1x,'nnoc =',i5,' nnorb=',i5,' nnvir =',i5,' numnb=',i5)
 9010 format(/1x,'hf  annihilation rate (1/nsec) =', f14.8/
     *        1x,'hf  positron lifetime (nsec  ) =', f14.8/
     *        1x,'mp2 ann rate corr     (1/nsec) =', f14.8/
     *        1x,'mp2 annihilation rate (1/nsec) =', f14.8/
     *        1x,'mp2 positron lifetime (nsec  ) =', f14.8/)
 9020 format(/1x,'hf  annihilation rate (1/nsec) =', f14.8/
     *        1x,'hf  positron lifetime (nsec  ) =', f14.8/
     *        1x,'fci annihilation rate (1/nsec) =', f14.8/
     *        1x,'fci positron lifetime (nsec  ) =', f14.8/)
       end
```

C.1.2 getlambdamp2 subroutine. The *getlambdamp2* subroutine, given in Listing C.2, calculates the NEO-HF and NEO-MP2 annihilation rates.

Listing C.2: *getlambdamp2* subroutine.

```
C*MODULE neopos  *DECK getlambdamp2
       subroutine getlambdamp2(is,it,xlhf,xlep,ce,cn,eige,eign,xx,ix,
     *   nintmx,cep,sep)
c
       implicit double precision(a-h,o-z)
c
       parameter (maxneo=40)
c
       common /iofile/ ir,iw,ip,ijk,ipk,idaf,nav,ioda(950)
       common /mp2par/ ospt,tol,method,nwdmp2,mpprop,nacore,
     *                 nbcore,noa,nob,no,nbf,nomit,mocphf,maxitc
       common /nucmoi/ nuniqn,iuniqn(maxneo),iuniqt(maxneo),nqmnuc,
     *                 iqmnuc(maxneo),iqntyp(maxneo),numnb,nucst,
     *                 nauxnb,iauxnb(maxneo),numult,nna,nnb,ntauxb
```

```
      common /pcklab/ labsiz
c
      dimension xx(nintmx),ix(nintmx*4)
      dimension cep(noa*no*numnb)
      dimension sep(noa*no*numnb)
      dimension ce(no*nbf),cn(numnb*numnb),eige(no),eign(numnb)
c
      noc=noa
      nnorb=numnb
c
      do i=1,noc*no*nnorb
        cep(i)=0.0d+00
        sep(i)=0.0d+00
      end do
c
c     get the EPA matrix and form the MP2 coefficients
c     EPA integrals stored in ⟨pp|ee⟩ order
c
      call seqrew(it)
  110 call pread(it,xx,ix,nxx,nintmx)
      if(nxx.eq.0) go to 180
      nint=iabs(nxx)
c
      do 150 m=1,nint
         val1 = xx(m)
         npack = m
c
c     ----- two bytes per label -----
c
         if(labsiz.ge.2) then
*32         label=ix(2*npack-1)
*32         ipack=ishft(label,-16)
*32         jpack=iand(label,65535)
*32         label=ix(2*npack)
*32         kpack=ishft(label,-16)
*32         lpack=iand(label,65535)
*64         label=ix(npack)
*64         ipack=ishft(label,-48)
*64         jpack=iand(ishft(label,-32),65535)
*64         kpack=iand(ishft(label,-16),65535)
*64         lpack=iand(label,65535)
         else
c
c     ----- one byte per label -----
c
*32         label=ix(npack)
*32         ipack=ishft(label,-24)
*32         jpack=iand(ishft(label,-16),255)
*32         kpack=iand(ishft(label,-8),255)
*32         lpack=iand(label,255)
*64         if (mod(npack,2).eq.0) then
*64            label=ix(npack/2)
```

```
*64            ipack=iand(ishft(label,-24),255)
*64            jpack=iand(ishft(label,-16),255)
*64            kpack=iand(ishft(label,-8),255)
*64            lpack=iand(label,255)
*64         else
*64            label=ix((npack/2)+1)
*64            ipack=ishft(label,-56)
*64            jpack=iand(ishft(label,-48),255)
*64            kpack=iand(ishft(label,-40),255)
*64            lpack=iand(ishft(label,-32),255)
*64         end if
          end if
          jip = ipack
          jjp = jpack
          ji = kpack
          jj = lpack
          jap=1
          do 140 ja=1,noc
             do 130 jr=noc+1,no
                do 120 jrp=2,nnorb
                   denom = eige(ja)+eign(jap)-eige(jr)-eign(jrp)
                   icep=(ja-1)*no*nnorb+(jr-1)*nnorb+jrp
                   ctmpa=cep(icep)
                   if ((ji.eq.jj).and.(jip.eq.jjp)) then
                     ctmp= val1
     *               * ce((ja -1)*nbf  +ji )
     *               * ce((jr -1)*nbf  +jj )
     *               * cn((jap-1)*numnb+jip)
     *               * cn((jrp-1)*numnb+jjp)
                   else if (ji.eq.jj) then
                     ctmp= val1
     *               *(ce((ja -1)*nbf  +ji )
     *               * ce((jr -1)*nbf  +jj ))
     *               *(cn((jap-1)*numnb+jip)
     *               * cn((jrp-1)*numnb+jjp)
     *               + cn((jap-1)*numnb+jjp)
     *               * cn((jrp-1)*numnb+jip))
                   else if (jip.eq.jjp) then
                     ctmp= val1
     *               *(cn((jap-1)*numnb+jip)
     *               * cn((jrp-1)*numnb+jjp))
     *               *(ce((ja -1)*nbf  +ji ) * ce((jr -1)*nbf  +jj )
     *               + ce((ja -1)*nbf  +jj ) * ce((jr -1)*nbf  +ji ))
                   else
                     ctmp= val1
     *               *(ce((ja -1)*nbf  +ji ) * ce((jr -1)*nbf  +jj )
     *               *(cn((jap-1)*numnb+jip) * cn((jrp-1)*numnb+jjp)
     *               + cn((jap-1)*numnb+jjp) * cn((jrp-1)*numnb+jip))
     *               + ce((ja -1)*nbf  +jj ) * ce((jr -1)*nbf  +ji )
     *               *(cn((jap-1)*numnb+jip) * cn((jrp-1)*numnb+jjp)
     *               + cn((jap-1)*numnb+jjp) * cn((jrp-1)*numnb+jip)))
                   end if
```

```
                  cep(icep) = ctmpa + ctmp
  120             continue
  130          continue
  140       continue
  150    continue
         if(nxx.gt.0) go to 110
  180 continue
c
c     compute the mp2 correction to the annihilation rate
c     ovrlpints are stored in ⟨ee|pp⟩ order
c
      xlhf=0.0d+00
      call seqrew(is)
  200 call pread(is,xx,ix,nxx,nintmx)
      if(nxx.eq.0) return
      nint=iabs(nxx)
      do 250 m=1,nint
         val1 = xx(m)
         npack = m
         if(labsiz.ge.2) then
c
c     ----- two bytes per label -----
c
*32         label=ix(2*npack-1)
*32         ipack=ishft(label,-16)
*32         jpack=iand(label,65535)
*32         label=ix(2*npack)
*32         kpack=ishft(label,-16)
*32         lpack=iand(label,65535)
*64         label=ix(npack)
*64         ipack=ishft(label,-48)
*64         jpack=iand(ishft(label,-32),65535)
*64         kpack=iand(ishft(label,-16),65535)
*64         lpack=iand(label,65535)
         else
c
c     ----- one byte per label -----
c
*32         label=ix(npack)
*32         ipack=ishft(label,-24)
*32         jpack=iand(ishft(label,-16),255)
*32         kpack=iand(ishft(label,-8),255)
*32         lpack=iand(label,255)
*64         if(mod(npack,2).eq.0) then
*64            label=ix(npack/2)
*64            ipack=iand(ishft(label,-24),255)
*64            jpack=iand(ishft(label,-16),255)
*64            kpack=iand(ishft(label,-8),255)
*64            lpack=iand(label,255)
*64         else
*64            label=ix((npack/2)+1)
*64            ipack=ishft(label,-56)
```

```
*64             jpack=iand(ishft(label,-48),255)
*64             kpack=iand(ishft(label,-40),255)
*64             lpack=iand(ishft(label,-32),255)
*64           end if
            end if
            ji = ipack
            jj = jpack
            jip = kpack
            jjp = lpack
            jap=1
            do 240 ja=1,noc
             do 230 jr=1,no
              do 220 jrp=1,nnorb
c
c               i = j and i' = j' so only need (ii|i'i')
c
                if ((ji.eq.jj).and.(jip.eq.jjp)) then
                   stmp= val1
     *             * ce((ja -1)*nbf  +ji ) * ce((jr -1)*nbf  +ji )
     *             * cn((jap-1)*numnb+jip) * cn((jrp-1)*numnb+jip)
c
c                  i = j so only need (ii|i'j') and (ii|j'i')
c
                else if (ji.eq.jj) then
                   stmp= val1
     *             *(ce((ja -1)*nbf  +ji ) * ce((jr -1)*nbf  +ji ))
     *             *(cn((jap-1)*numnb+jip) * cn((jrp-1)*numnb+jjp)
     *             + cn((jap-1)*numnb+jjp) * cn((jrp-1)*numnb+jip))
c
c                  i' = j' so only need (ij|i'i') and (ji|i'i')
c
                else if (jip.eq.jjp) then
                   stmp= val1
     *             *(cn((jap-1)*numnb+jip) * cn((jrp-1)*numnb+jip))
     *             *(ce((ja -1)*nbf  +ji ) * ce((jr -1)*nbf  +jj )
     *             + ce((ja -1)*nbf  +jj ) * ce((jr -1)*nbf  +ji ))
c
c                  need all (ij|i'j'), (ij|j'i'), (ji|i'j'), and (ji|j'i')
c
                else
                   stmp= val1
                   *(ce((ja -1)*nbf  +ji ) * ce((jr -1)*nbf  +jj )
                   *(cn((jap-1)*numnb+jip) * cn((jrp-1)*numnb+jjp)
                   + cn((jap-1)*numnb+jjp) * cn((jrp-1)*numnb+jip))
                   + ce((ja -1)*nbf  +jj ) * ce((jr -1)*nbf  +ji )
                   *(cn((jap-1)*numnb+jip) * cn((jrp-1)*numnb+jjp)
                   + cn((jap-1)*numnb+jjp) * cn((jrp-1)*numnb+jip)))
                end if
c
                isep=(ja-1)*no*nnorb+(jr-1)*nnorb+jrp
                stmpb=sep(isep)
                sep(isep)= stmp + stmpb
```

```
c
c             we can compute xlhf here or later
c
              if((ja.eq.jr).and.(jrp.eq.1)) then
                 xlhf=xlhf+ 2.0d+00*stmp
              end if
  220       continue
  230      continue
  240     continue
c
  250 continue
c
      if(nxx.gt.0) go to 200
      xlhf=0.0d+00
      do ja=1,noa
         isep=(ja-1)*no*nnorb +(ja-1)*nnorb+1
         xlhf=xlhf+ 2.0d+00*sep(isep)
      end do
      xlep=0.0d+00
      xeep=0.0d+00
      jap=1
      do 340 ja=1,noc
         do 330 jr=noc+1,no
            do 320 jrp=2,nnorb
               denom = eige(ja)+eign(jap)-eige(jr)-eign(jrp)
               isep=(ja-1)*no*nnorb+(jr-1)*nnorb+jrp
               xlep=xlep + 2.0d+00*sep(isep)*cep(isep)/denom
               xeep=xeep + 2.0d+00*cep(isep)*cep(isep)/denom
  320       continue
  330    continue
  340 continue
      write(iw,9000) xeep
c
      return
 9000 format(/1x,'recalculate e(2ep) as a check: ', f14.8/)
      end
```

C.1.3 neodftint subroutine. The *neodftint* subroutine, given in Listing C.3, calculates the four-center integrals for use in the annihilation rate codes.

Listing C.3: *neodftint* subroutine.

```
c*MODULE neopos  *DECK neodftint
      subroutine neodftint(idftovrlp)
      implicit double precision(a-h,o-z)
c
      logical dirscf,fdiff,pack2e,some,goparr,dskwrk,maswrk
c
      parameter (mxatm=2000)
c
```

```
      common /dftpar/ dfttyp(20),exena,exenb,exenc,idft34,nauxfun,
     *                nauxshl
      common /fmcom / x(1)
      common /infoa / nat,ich,mul,num,nqmt,ne,na,nb,
     *                zan(mxatm),c(3,mxatm),ian(mxatm)
      common /intfil/ nintmx,nhex,ntupl,pack2e,inttyp,igrdtyp
      common /iofile/ ir,iw,ip,is,ipk,idaf,nav,ioda(950)
      common /neoshl/ ngauss,ngause,ngausn,ntshel,nnucsh
      common /optscf/ dirscf,fdiff
      common /output/ nprint,itol,icut,normf,normp,nopk
      common /par   / me,master,nproc,ibtyp,iptim,goparr,dskwrk,
     *                maswrk
      common /runopt/ runtyp,exetyp,nevals,nglevl,nhlevl
c
      data check  /8hcheck   /
c
      some = maswrk  .and.  nprint.ne.-5
      if(some) write(iw,9000)
      l1 = num
      l2 = (num*(num+1))/2
c
      if(idftovrlp.ne.4) then
         write(iw,*) 'idftovrlp confused in neodftint'
         call abrt
      end if
c
c     do four-center integrals for neopos
c     adapted from grid-free dft method
c
      if(idftovrlp .eq. 4) then
         if(.not. dirscf) then
            call valfm(loadfm)
            iscrn = loadfm + 1
            itmp  = iscrn  + (ntshel*ntshel+ntshel)/2
            iss   = itmp   + 15*15*15*15
            iloc  = iss    + nintmx
            last  = iloc   + nintmx
            need  = last - loadfm - 1
            call getfm(need)
            if(exetyp.ne.check) then
               call neoforscr(some,x(iscrn),x(itmp))
               call neofourc(some,x(iss),x(iloc),x(itmp),nintmx)
            end if
            call retfm(need)
         else
            call valfm(loadfm)
            iscrn=loadfm+1
            itmp=iscrn+(ntshel*ntshel+ntshel)/2
            last=itmp+15*15*15*15
            need = last - loadfm - 1
            call getfm(need)
            if(exetyp.ne.check) call neoforscr(some,x(iscrn),x(itmp))
```

```
            call retfm(need)
         end if
      end if
      if(maswrk) write(iw,9050)
      return
c
 9000 format(/10x,45(1h-)/10x,
     *   'electron-positron overlap integral evaluation'/10x,45(1h-))
 9030 format(1x,'time to evaluate gradient correction integrals=',f12...
     .1)
 9050 format(1x,'..... end of electron-positron overlap integrals...
     .....')
      end
```

C.1.4 neofourc subroutine. The *neofourc* subroutine, given in Listing C.4, is used by *neodftint* for calculating the four-center integrals.

Listing C.4: *neofourc* subroutine.

```
C*MODULE neopos  *DECK neofourc
      subroutine neofourc(some,s,ix,temps,nintmx)
      implicit double precision(a-h,o-z)
c
      logical samecen,svdskw,some,dskwrk,maswrk,goparr
c
      parameter (mxsh=5000, mxgtot=20000, mxatm=2000)
c
      common /iofile/ ir,iw,ip,ijko,ijkt,idaf,nav,ioda(950)
      common /infoa / nat,ich,mul,num,nqmt,ne,na,nb,
     *                zan(mxatm),c(3,mxatm),ian(mxatm)
      common /neoshl/ ngauss,ngause,ngausn,ntshel,nnucsh
      common /nshel / ex(mxgtot),cs(mxgtot),cp(mxgtot),cd(mxgtot),
     *                cf(mxgtot),cg(mxgtot),ch(mxgtot),ci(mxgtot),
     *                kstart(mxsh),katom(mxsh),ktype(mxsh),kng(mxsh),
     *                kloc(mxsh),kmin(mxsh),kmax(mxsh),nshell
      common /output/ nprint,itol,icut,normf,normp,nopk
      common /par   / me,master,nproc,ibtyp,iptim,goparr,dskwrk,
     *                maswrk
      common /pcklab/ labsiz
      common /symtry/ mapshl(mxsh,48),mapctr(mxatm,48),
     *                t(432),invt(48),nt
      common /restar/ timlim,irest,nrec,intloc,ist,jst,kst,lst
c
      dimension s(nintmx),ix(*),temps(15*15*15*15)
      dimension dfac(0:21),ldat(35),mdat(35),ndat(35),
     *          conta(35),contb(35),contc(35),conte(35),
     *          a21(0:4*4*4*4),idummy(0:4),mi(48),mj(48),mk(48),
     *          xxxx(-1:4,-1:4,-1:4,-1:4),
     *          yyyy(-1:4,-1:4,-1:4,-1:4),
     *          zzzz(-1:4,-1:4,-1:4,-1:4)
```

```
c
      parameter (pi=3.14159265358979323846264338328d+00,
     *           rln10=2.30258d+00)
c
c     x,y,z exponents of cartesian s,px,py,pz,dxx,...
c
      data ldat/0,1,0,0,2,0,0,1,1,0,3,0,0,2,2,1,0,1,0,1,4,0,0,3,3,
     *          1,0,1,0,2,2,0,2,1,1/
      data mdat/0,0,1,0,0,2,0,1,0,1,0,3,0,1,0,2,2,0,1,1,0,4,0,1,0,
     *          3,3,0,1,2,0,2,1,2,1/
      data ndat/0,0,0,1,0,0,2,0,1,1,0,0,3,0,1,0,1,2,2,1,0,0,4,0,1,
     *          0,1,3,3,0,2,2,1,1,2/
c
c     calculates ivalue!!
c     warning dfac(0) = -1!!, dfac(1) = 0!!, dfac(2) = 1!! etc.
c
      data dfac/1.0d+00,1.0d+00,1.0d+00,2.0d+00,3.0d+00,8.0d+00,
     *          15.0d+00,48.0d+00,105.0d+00,384.0d+00,945.0d+00,
     *          3840.0d+00,10395.0d+00,46080.0d+00,135135.0d+00,
     *          645120.0d+00,2027025.0d+00,10321920.0d+00,
     *          34459425.0d+00,185794560.0d+00,
     *          654729075.0d+00,3715891200.0d+00/
c
c     helps deal with l shells
c     (shell that ends on 1 might start with 0)
c
      data idummy/0,0,2,3,4/
c
c     this routine calculates the one electron four-overlap integral
c     for cartesian gaussians.  (up to g-shells)
c
      if ((normf .eq. 1 .or. normp .eq. 1).and. maswrk) then
         write(iw,*)' sorry but the four center one electron ovelap'
         write(iw,*)' integrals used for dft do not support the'
         write(iw,*)' no normalization options'
         call abrt
      end if
c
c     ----- initialization for parallel work -----
c
      ipcount = me - 1
      svdskw = dskwrk
      dskwrk = .true.
c
      if(some) write(iw,9000)
c
      call tsecnd(tim0)
c
      cutoff = 1.0d+00/(10.0d+00**icut)
      tol=itol*rln10
      iwhere=0
      call vclr(xxxx,1,6*6*6*6)
```

```
      call vclr(yyyy,1,6*6*6*6)
      call vclr(zzzz,1,6*6*6*6)
      xxxx(0,0,0,0)=1.0d+00
      yyyy(0,0,0,0)=1.0d+00
      zzzz(0,0,0,0)=1.0d+00
c
      nschwz=0
      nrec=0
      nints=0
      maxao=255
      labsiz = 1
      if(num.gt.maxao) labsiz = 2
      call seqopn(21,'dftints','unknown',.false.,'unformatted')
      call seqrew(21)
c
c     loop over shells a b c and e for ⟨ab|ce⟩
c
      intshel=nshell+1
      iashl=1
      ibshl=1
      icshl=intshel
      ieshl=intshel
      do 100 iashl=1,nshell
c
c     ----- check cpu time -----
c
         call tsecnd(tim)
         if(tim.ge.timlim) then
            write(iw,*) 'no time left in -fourc-'
            return
         end if
c
c        try to eliminate a shell
c
         do 1020 it = 1,nt
            id = mapshl(iashl,it)
            if (id .gt. iashl) go to 100
            mi(it) = id
 1020    continue
c
c        position of the shell's atom
c
         iaatom=katom(iashl)
         ax=c(1,iaatom)
         ay=c(2,iaatom)
         az=c(3,iaatom)
c
c        info about the primatives
c
         istarta=kstart(iashl)
         ienda=istarta+kng(iashl)-1
c
```

```
c         info about the subshells
c
          iamax=kmax(iashl)
          iamin=kmin(iashl)
          loca =kloc(iashl)-iamin
c
c         find the highest spin value in this shell
c
          katype=ktype(iashl)-1
          jatype=1+iamax-iamin
c
c         b shell
c
          do 101 ibshl=1,nshell
             do 200 it = 1,nt
                id = mi(it)
                jd = mapshl(ibshl,it)
                mj(it) = jd
                if (id .ge. jd) go to 1060
                nd = id
                id = jd
                jd = nd
 1060           if (id-iashl) 200,1080,101
 1080           if (jd-ibshl) 200,200,101
  200        continue
c
             ibatom=katom(ibshl)
             bx=c(1,ibatom)
             by=c(2,ibatom)
             bz=c(3,ibatom)
             istartb=kstart(ibshl)
             iendb=istartb+kng(ibshl)-1
             ibmax=kmax(ibshl)
             ibmin=kmin(ibshl)
             locb = kloc(ibshl)-ibmin
             kbtype=ktype(ibshl)-1
             jbtype=1+ibmax-ibmin
             abrr=(ax-bx)*(ax-bx)+(ay-by)*(ay-by)+(az-bz)*(az-bz)
c
c            c shell
c

             do 102 icshl=intshel,ntshel
                do 340 it = 1,nt
                   id = mi(it)
                   jd = mj(it)
                   kd = mapshl(icshl,it)
                   mk(it) = kd
                   if (id .ge. jd) go to 260
                   nd = id
                   id = jd
                   jd = nd
```

```
  260          continue
               if (id-iashl) 340,300,102
  300          if (jd-ibshl) 340,320,102
  320          if (kd-icshl) 340,340,102
  340       continue
            icatom=katom(icshl)
            cx=c(1,icatom)
            cy=c(2,icatom)
            cz=c(3,icatom)
            istartc=kstart(icshl)
            iendc=istartc+kng(icshl)-1
            icmax=kmax(icshl)
            icmin=kmin(icshl)
            locc = kloc(icshl)-icmin
            kctype=ktype(icshl)-1
            jctype=1+icmax-icmin
            acrr=(ax-cx)*(ax-cx)+(ay-cy)*(ay-cy)+(az-cz)*(az-cz)
            bcrr=(bx-cx)*(bx-cx)+(by-cy)*(by-cy)+(bz-cz)*(bz-cz)
c
c           e shell
c
            do 103 ieshl=intshel,ntshel
               n4 = 0
               do 540 it = 1,nt
                  id = mi(it)
                  jd = mj(it)
                  kd = mk(it)
                  ld = mapshl(ieshl,it)
                  if (id .ge. jd) go to 400
                  nd = id
                  id = jd
                  jd = nd
  400             continue
                  if (kd .ge. ld) go to 440
                  nd = kd
                  kd = ld
                  ld = nd
                  go to 400
  440             if (id-iashl) 540,460,103
  460             if (jd-ibshl) 540,480,103
  480             if (kd-icshl) 540,500,103
  500             if (ld-ieshl) 540,520,103
  520             n4 = n4+1
  540       continue
            q4=nt
            q4=q4/n4
c
c     ----- go parallel! -----
c
            if (goparr) then
               ipcount = ipcount + 1
               if (mod(ipcount,nproc).ne.0) go to 103
```

```
            end if
c
            ieatom=katom(ieshl)
            exx=c(1,ieatom)
            ey=c(2,ieatom)
            ez=c(3,ieatom)
            istarte=kstart(ieshl)
            iende=istarte+kng(ieshl)-1
            iemax=kmax(ieshl)
            iemin=kmin(ieshl)
            loce = kloc(ieshl)-iemin
            ketype=ktype(ieshl)-1
            jetype=1+iemax-iemin
            aerr=(ax-exx)*(ax-exx)+(ay-ey)*(ay-ey)+(az-ez)*(az-ez)
            berr=(bx-exx)*(bx-exx)+(by-ey)*(by-ey)+(bz-ez)*(bz-ez)
            cerr=(cx-exx)*(cx-exx)+(cy-ey)*(cy-ey)+(cz-ez)*(cz-ez)
            if ((aerr.gt.cutoff).or.(berr.gt.cutoff).or.
     *         (cerr.gt.cutoff).or.(acrr.gt.cutoff).or.
     *         (bcrr.gt.cutoff).or.(abrr.gt.cutoff)) then
               samecen=.false.
            else
               samecen=.true.
            end if
c
c           end of shells
c
c           disable schwartz screening
c           pre-screen the integrals
c
c           if(smax(kashl+ibshl)*smax(kcshl+ieshl) .le. 1.0d-18 .or.
c    *         smax(kashl+icshl)*smax(kbshl+ieshl) .le. 1.0d-18 .or.
c    *         smax(kashl+ieshl)*smax(kbshl+icshl) .le. 1.0d-18) then
c              nschwz=nschwz+1
c              go to 103
c           end if
c
c           check to see if they are all on the same center
c
c           if(iaatom .eq. ibatom .and. iaatom .eq. icatom .and.
c              iaatom .eq. ieatom) then
c
            if (samecen) then
               if(  mod(katype+kbtype+kctype+ketype,2).eq.1 .and.
     *            .not.(iamin.eq.1 .and. iamax.eq.4).and.
     *            .not.(ibmin.eq.1 .and. ibmax.eq.4).and.
     *            .not.(icmin.eq.1 .and. icmax.eq.4).and.
     *            .not.(iemin.eq.1 .and. iemax.eq.4)) go to 103
c
c              no l shells to confuse things, and total
c              momentum is odd, thus integrals are zero
c
c              samecen=.true.
```

```
c           else
c              samecen=.false.
c
            end if
c
c           loop over guassian primatives while reading
c           in exponents and contractions
c
c           do something special for (ssss) case
c
            if(katype.eq.0 .and. kbtype.eq.0 .and. kctype.eq.0 .and.
     *         ketype.eq.0) then
               temps(1)=0.0d+00
               if (samecen) then
                  do 910 iaprm=istarta,ienda
                     a1=ex(iaprm)
                     sconta=cs(iaprm)
                     do 911 ibprm=istartb,iendb
                        a2=ex(ibprm)+a1
                        scontb=cs(ibprm)*sconta
                        do 912 icprm=istartc,iendc
                           a3=ex(icprm)+a2
                           scontc=cs(icprm)*scontb
                           do 913 ieprm=istarte,iende
                              a5=ex(ieprm)+a3
                              sconte=cs(ieprm)*scontc
                              temps(1)=temps(1)+sconte/(a5*sqrt(a5))
  913                      continue
  912                   continue
  911                continue
  910             continue
                  temps(1)=temps(1)*pi*sqrt(pi)
               else
               do 810 iaprm=istarta,ienda
                  a1=ex(iaprm)
                  sconta=cs(iaprm)
                  do 811 ibprm=istartb,iendb
                     a2=ex(ibprm)
                     if(abrr*a1*a2/(a1+a2) .gt. tol) go to 811
                     scontb=cs(ibprm)*sconta
c
c                    get info about a,b,and p
c
                     a6=a1+a2
                     px=((a1*ax+a2*bx)/a6)
                     py=((a1*ay+a2*by)/a6)
                     pz=((a1*az+a2*bz)/a6)
                     abk=-((a1*a2/a6)*((ax-bx)*(ax-bx)+(ay-by)*
     *                  (ay-by)+(az-bz)*(az-bz)))
                     do 812 icprm=istartc,iendc
                        a3=ex(icprm)
                        if(acrr*a1*a3/(a1+a3) .gt. tol .or.
```

```
     *                     bcrr*a2*a3/(a2+a3) .gt. tol) go to 812
                        scontc=cs(icprm)*scontb
                        do 813 ieprm=istarte,iende
                           a5=ex(ieprm)
                           if(aerr*a1*a5/(a1+a5) .gt. tol .or.
     *                        berr*a2*a5/(a2+a5) .gt. tol .or.
     *                        cerr*a3*a5/(a3+a5) .gt. tol) go to 813
                           sconte=cs(ieprm)*scontc
                           a4=a1+a2+a3+a5
                           a4inv=1.0d+00/a4
                           a7=a3+a5
                           a7inv=1.0d+00/a7
                           qx=(a3*cx+a5*exx)*a7inv
                           qy=(a3*cy+a5*ey)*a7inv
                           qz=(a3*cz+a5*ez)*a7inv
c
c                          calculate
c
                           temps(1)=temps(1)+sconte*(((pi*a4inv)*
     *                        sqrt(pi*a4inv))*exp(abk-((a3*a5*a7inv)*
     *                        ((cx-exx)*(cx-exx)+(cy-ey)*
     *                         (cy-ey)+(cz-ez)*(cz-ez)))
     *                        -((a6*a7*a4inv)*
     *                        ((px-qx)*(px-qx)+(py-qy)*
     *                         (py-qy)+(pz-qz)*(pz-qz)))))
  813                   continue
  812                continue
  811             continue
  810          continue
            end if
c
c           end special (ssss) case
c
         else
            call vclr(temps,1,jatype*jbtype*jctype*jetype)
            do 110 iaprm=istarta,ienda
               a1=ex(iaprm)
               call forcnt(iaprm,iamax,conta)
               do 111 ibprm=istartb,iendb
                  a2=ex(ibprm)
                  call forcnt(ibprm,ibmax,contb)
c
c                 get info about a,b,and p
c
                  if(.not. samecen) then
                     a6=a1+a2
                     px=((a1*ax+a2*bx)/a6)
                     py=((a1*ay+a2*by)/a6)
                     pz=((a1*az+a2*bz)/a6)
                     abk=(a1*a2/a6)*abrr
                     if (abk .gt. tol) go to 111
                  end if
```

```
                  do 112 icprm=istartc,iendc
                     a3=ex(icprm)
                     call forcnt(icprm,icmax,contc)
                     if(acrr*a1*a3/(a1+a3) .gt. tol .or.
     *                  bcrr*a2*a3/(a2+a3) .gt. tol) go to 112
                     do 113 ieprm=istarte,iende
                     a5=ex(ieprm)
                     call forcnt(ieprm,iemax,conte)
c
c    calculate final gaussian center and other things
c    p=a and b, q=c and e, d=final center
c
                     a4=a1+a2+a3+a5
                     a4inv=1.0d+00/a4
                     if(.not. samecen) then
                        if(aerr*a1*a5/(a1+a5) .gt. tol .or.
     *                     berr*a2*a5/(a2+a5) .gt. tol .or.
     *                     cerr*a3*a5/(a3+a5) .gt. tol) go to 113
                        a7=a3+a5
                        a7inv=1.0d+00/a7
                        a10=0.5d+00*a4inv
                        qx=(a3*cx+a5*exx)*a7inv
                        qy=(a3*cy+a5*ey)*a7inv
                        qz=(a3*cz+a5*ez)*a7inv
                        dx=(a1*ax+a2*bx+a3*cx+a5*exx)*a4inv
                        dy=(a1*ay+a2*by+a3*cy+a5*ey)*a4inv
                        dz=(a1*az+a2*bz+a3*cz+a5*ez)*a4inv
                        dax=dx-ax
                        day=dy-ay
                        daz=dz-az
                        dbx=dx-bx
                        dby=dy-by
                        dbz=dz-bz
                        dcx=dx-cx
                        dcy=dy-cy
                        dcz=dz-cz
                        dex=dx-exx
                        dey=dy-ey
                        dez=dz-ez
                        pqrr=(px-qx)*(px-qx)+(py-qy)*(py-qy)+
     *                       (pz-qz)*(pz-qz)
c
c                       calculate kabc
c
                        abck=((pi*a4inv)*sqrt(pi*a4inv))*exp(-abk
     *                     -((a3*a5*a7inv)*cerr)
     *                     -((a6*a7*a4inv)*pqrr))
c
c     use recursion to generate higher angular momentum xxxx,etc
c
c     recursion based on j
c
```

```
                      do 2000 j=1,ketype
                  xxxx(j,0,0,0)= dex*xxxx(j-1,0  ,0  ,0  )+a10*(
     *                          (j-1)*xxxx(j-2,0  ,0  ,0  ))
                  yyyy(j,0,0,0)= dey*yyyy(j-1,0  ,0  ,0  )+a10*(
     *                          (j-1)*yyyy(j-2,0  ,0  ,0  ))
                  zzzz(j,0,0,0)= dez*zzzz(j-1,0  ,0  ,0  )+a10*(
     *                          (j-1)*zzzz(j-2,0  ,0  ,0  ))
 2000                 continue
c     recursion based on k
                      do 2010 k=1,kctype
                         do 2010 j=0,ketype
                  xxxx(j,k,0,0)= dcx*xxxx(j  ,k-1,0  ,0  )+a10*(
     *                             j*xxxx(j-1,k-1,0  ,0  ) +
     *                          (k-1)*xxxx(j  ,k-2,0  ,0  ))
                  yyyy(j,k,0,0)= dcy*yyyy(j  ,k-1,0  ,0  )+a10*(
     *                             j*yyyy(j-1,k-1,0  ,0  ) +
     *                          (k-1)*yyyy(j  ,k-2,0  ,0  ))
                  zzzz(j,k,0,0)= dcz*zzzz(j  ,k-1,0  ,0  )+a10*(
     *                             j*zzzz(j-1,k-1,0  ,0  ) +
     *                          (k-1)*zzzz(j  ,k-2,0  ,0  ))
 2010                    continue
c     recursion based on l
                      do 2020 l=1,kbtype
                         do 2020 k=0,kctype
                            do 2020 j=0,ketype
                  xxxx(j,k,l,0)= dbx*xxxx(j  ,k  ,l-1,0  )+a10*(
     *                             j*xxxx(j-1,k  ,l-1,0  ) +
     *                             k*xxxx(j  ,k-1,l-1,0  ) +
     *                          (l-1)*xxxx(j  ,k  ,l-2,0  ))
                  yyyy(j,k,l,0)= dby*yyyy(j  ,k  ,l-1,0  )+a10*(
     *                             j*yyyy(j-1,k  ,l-1,0  ) +
     *                             k*yyyy(j  ,k-1,l-1,0  ) +
     *                          (l-1)*yyyy(j  ,k  ,l-2,0  ))
                  zzzz(j,k,l,0)= dbz*zzzz(j  ,k  ,l-1,0  )+a10*(
     *                             j*zzzz(j-1,k  ,l-1,0  ) +
     *                             k*zzzz(j  ,k-1,l-1,0  ) +
     *                          (l-1)*zzzz(j  ,k  ,l-2,0  ))
 2020                       continue
c     recursion based on i
                      do 2030 i=1,katype
                         do 2030 l=0,kbtype
                            do 2030 k=0,kctype
                               do 2030 j=0,ketype
                  xxxx(j,k,l,i)= dax*xxxx(j  ,k  ,l  ,i-1)+a10*(
     *                          (i-1)*xxxx(j  ,k  ,l  ,i-2) +
     *                              j*xxxx(j-1,k  ,l  ,i-1) +
     *                              k*xxxx(j  ,k-1,l  ,i-1) +
     *                              l*xxxx(j  ,k  ,l-1,i-1))
                  yyyy(j,k,l,i)= day*yyyy(j  ,k  ,l  ,i-1)+a10*(
     *                          (i-1)*yyyy(j  ,k  ,l  ,i-2) +
     *                              j*yyyy(j-1,k  ,l  ,i-1) +
     *                              k*yyyy(j  ,k-1,l  ,i-1) +
```

```
     *                              l*yyyy(j  ,k  ,l-1,i-1))
                    zzzz(j,k,l,i)= daz*zzzz(j  ,k  ,l  ,i-1)+a10*(
     *                            (i-1)*zzzz(j  ,k  ,l  ,i-2) +
     *                                j*zzzz(j-1,k  ,l  ,i-1) +
     *                                k*zzzz(j  ,k-1,l  ,i-1) +
     *                                l*zzzz(j  ,k  ,l-1,i-1))
 2030                             continue
c
c     loop over subshells to get values of the integrals
c
          imove=0
          do 120 iasub=iamin,iamax
c          the x coordinate in s(num,num,num,num)
           iaposit=loca+iasub
           l1=ldat(iasub)
           m1=mdat(iasub)
           n1=ndat(iasub)
           contra=abck*conta(iasub)
c
           do 121 ibsub=ibmin,ibmax
c           the y coordinate in s(num,num,num,num)
            ibposit=locb+ibsub
            if(ibposit .gt. iaposit) go to 120
            l2=ldat(ibsub)
            m2=mdat(ibsub)
            n2=ndat(ibsub)
            contrb=contra*contb(ibsub)
c
            do 122 icsub=icmin,icmax
c            the z coordinate in s(num,num,num,num)
             icposit=locc+icsub
             l3=ldat(icsub)
             m3=mdat(icsub)
             n3=ndat(icsub)
             contrc=contrb*contc(icsub)
c
            do 123 iesub=iemin,iemax
c            the e coordinate in s(num,num,num,num)
             ieposit=loce+iesub
             if(ieposit .gt. icposit) go to 122
             l5=ldat(iesub)
             m5=mdat(iesub)
             n5=ndat(iesub)
             contre=contrc*conte(iesub)
             imove=imove+1
c
             temps(imove) = temps(imove)
     *        +contre*xxxx(l5,l3,l2,l1)
     *        *yyyy(m5,m3,m2,m1)*zzzz(n5,n3,n2,n1)
c
  123       continue
  122       continue
```

```
  121      continue
  120     continue
          call flshbf(iw)
          else
c
c         special same center case
c
          imove=0
          radial=(pi*a4inv)*sqrt(pi*a4inv)
          a20=sqrt(0.5d+00*a4inv)
          idum1=idummy(katype)+idummy(kbtype)+idummy(kctype)+
     *     idummy(ketype)
          idum2=katype+kbtype+kctype+ketype
          do i=idum1,idum2
           a21(i)=a20**i
          end do
          do 220 iasub=iamin,iamax
c
c          the x coordinate in s(num,num,num,num)
c
           iaposit=loca+iasub
           l1=ldat(iasub)
           m1=mdat(iasub)
           n1=ndat(iasub)
           contra=conta(iasub)*radial
           do 221 ibsub=ibmin,ibmax
c
c           the y coordinate in s(num,num,num,num)
c
            ibposit=locb+ibsub
            if(ibposit .gt. iaposit) go to 220
            l2=l1+ldat(ibsub)
            m2=m1+mdat(ibsub)
            n2=n1+ndat(ibsub)
            contrb=contra*contb(ibsub)
            do 222 icsub=icmin,icmax
c
c            the z coordinate in s(num,num,num,num)
c
             icposit=locc+icsub
c            if(icposit .gt. ibposit) go to 221
             l3=l2+ldat(icsub)
             m3=m2+mdat(icsub)
             n3=n2+ndat(icsub)
             contrc=contrb*contc(icsub)
            do 223 iesub=iemin,iemax
c
c            the e coordinate in s(num,num,num,num)
c
             ieposit=loce+iesub
             if(ieposit .gt. icposit) go to 222
             imove=imove+1
```

```
              l5=l3+ldat(iesub)
              if(mod(l5,2) .eq. 1) go to 223
              m5=m3+mdat(iesub)
              if(mod(m5,2) .eq. 1) go to 223
              n5=n3+ndat(iesub)
              if(mod(n5,2) .eq. 1) go to 223
              contre=contrc*conte(iesub)
c
c             the if mod(n5,2) gets rid of odd exponents
c
             xyzi=dfac(l5)*dfac(m5)*dfac(n5)*contre*a21(l5+n5+m5)
             temps(imove)=temps(imove)+xyzi
  223        continue
  222        continue
  221       continue
  220      continue
c
c         end special same center case
c
          end if
c
  113     continue
  112     continue
  111    continue
  110   continue
c
c       end the ssss if
c
        end if
c
c       loop over temporary matrix
c
          imove=0
          do 421 iaposit=iamin+loca,iamax+loca
c
c           min lets us skip (5611) since we do (6511)
c
           do 422 ibposit=ibmin+locb,min(ibmax+locb,iaposit)
            do 423 icposit=icmin+locc,icmax+locc
             do 424 ieposit=iemin+loce,min(iemax+loce,icposit)
              imove=imove+1
c
c       postscreen the integrals
c
        if(abs(temps(imove)) .gt. cutoff) then
         iwhere=iwhere+1
         npack = iwhere
         ipack = iaposit
         jpack = ibposit
         kpack = icposit
         lpack = ieposit
         if (labsiz .eq. 2) then
```

```
*32       label1=ishft(ipack,16)+jpack
*32       label2=ishft(kpack,16)+lpack
*32       ix(2*npack-1)=label1
*32       ix(2*npack)=label2
*64       label=ishft(ipack,48)+ishft(jpack,32)+ishft(kpack,16)+lpack
*64       ix(npack)=label
         else if (labsiz .eq. 1) then
*32       label=ishft(ipack,24)+ishft(jpack,16)+ishft(kpack,8)+lpack
*32       ix(npack)=label
*64       if ( mod(npack,2).eq.0) then
*64        label=ishft(ipack,24)+ishft(jpack,16)+ishft(kpack,8)+lpack
*64        ix(npack/2)=ix(npack/2)+label
*64       else
*64        label=ishft(ipack,56)+ishft(jpack,48)+
*64   *     ishft(kpack,40)+ishft(lpack,32)
*64        ix((npack/2)+1)=label
*64       end if
         end if
c
c              take into account skipped intergrals and store
c
               s(iwhere)=temps(imove)*q4
               ithisone=0
               if (ithisone.eq.1) then
                write(iw,*)'i j k l ovrlpint',ipack,jpack,kpack,
     *           lpack,s(iwhere)
               end if
c
c               write out buffer of integrals if its full
c
               if(iwhere .eq. nintmx) then
                call pwrit(21,s,ix,nintmx,nintmx)
                iwhere=0
                nrec=nrec+1
                nints=nints+nintmx
               end if

             end if
  424       continue
  423      continue
  422     continue
  421    continue
c
  103 continue
  102 continue
  101 continue
  100 continue
c
      call flshbf(iw)
c
c     ----- end of shell loops -----
```

```
c
c     write final partial buffer load of integrals to disk
c
      nrec=nrec+1
      nints=nints+iwhere
      iwhere=-iwhere
      call pwrit(21,s,ix,iwhere,nintmx)
c
      if(goparr) then
         call ddi_gsumi(1055,nschwz,1)
         call ddi_gsumi(1056,nints,1)
         call ddi_gsumi(1057,nrecs,1)
      end if
      if(some) write(iw,9010) nints,nrec,21
      if(some) write(iw,9020) nschwz
      call tsecnd(tim)
      tim = tim-tim0
      if(some) write(iw,9030) tim
      dskwrk = svdskw
      return
c
 9000 format(1x,'computing four center overlap integrals (ijkl)')
 9010 format(1x,'total number of nonzero overlap integrals=',i20/
     *       1x,i10,' integral records stored on disk file',i3,'.')
 9020 format(1x,'schwarz inequality test skipped',i10,
     *    ' integral blocks.')
 9030 format(1x,'time to evaluate (ijkl) overlap integrals =',f12.2)
      end
```

C.1.5 neoforscr subroutine. The *neoforscr* subroutine, given in Listing C.5, is used by *neodftint* for calculating the four-overlap integrals.

Listing C.5: *neoforscr* subroutine.

```
c*MODULE neopos  *DECK neoforscr
      subroutine neoforscr(some,s,temps)
c
      implicit double precision(a-h,o-z)
c
      parameter (mxsh=5000, mxgtot=20000, mxatm=2000, mxao=8192)
c
      logical goparr,dskwrk,maswrk,some
c
      common /par   / me,master,nproc,ibtyp,iptim,goparr,dskwrk,
     *                maswrk
      common /infoa / nat,ich,mul,num,nqmt,ne,na,nb,
     *                zan(mxatm),c(3,mxatm),ian(mxatm)
      common /iofile/ ir,iw,ip,ijko,ijkt,idaf,nav,ioda(950)
      common /neoshl/ ngauss,ngause,ngausn,ntshel,nnucsh
      common /nshel / ex(mxgtot),cs(mxgtot),cp(mxgtot),cd(mxgtot),
```

```
     *                cf(mxgtot),cg(mxgtot),ch(mxgtot),ci(mxgtot),
     *                kstart(mxsh),katom(mxsh),ktype(mxsh),kng(mxsh),
     *                kloc(mxsh),kmin(mxsh),kmax(mxsh),nshell
      common /output/ nprint,itol,icut,normf,normp,nopk
      common /ijpair/ index2(mxao)
c
      dimension s((ntshel*ntshel+ntshel)/2),temps(15*15*15*15)
      dimension ldat(35),mdat(35),ndat(35),
     *          conta(35),contb(35),contc(35),conte(35),
     *          xxxx(-1:4,-1:4,-1:4,-1:4),
     *          yyyy(-1:4,-1:4,-1:4,-1:4),
     *          zzzz(-1:4,-1:4,-1:4,-1:4)
c
      parameter (pi=3.14159265358979323846264338327950288d+00)
      parameter (rln10=2.30258d+00)
c
c     x,y,z exponents of cartesian s,px,py,pz,dxx,...
c
      data ldat/0,1,0,0,2,0,0,1,1,0,3,0,0,2,2,1,0,1,0,1,4,0,0,3,3,
     *          1,0,1,0,2,2,0,2,1,1/
      data mdat/0,0,1,0,0,2,0,1,0,1,0,3,0,1,0,2,2,0,1,1,0,4,0,1,0,
     *          3,3,0,1,2,0,2,1,2,1/
      data ndat/0,0,0,1,0,0,2,0,1,1,0,0,3,0,1,0,1,2,2,1,0,0,4,0,1,
     *          0,1,3,3,0,2,2,1,1,2/
c
      tol=itol*rln10
c
c     ----- intialize parallel -----
c
      ipcount = me - 1
      if (goparr) call vclr(s,1,ntshel*(ntshel+1)/2)
c
      call tsecnd(tim0)
c
      call vclr(xxxx,1,6*6*6*6)
      call vclr(yyyy,1,6*6*6*6)
      call vclr(zzzz,1,6*6*6*6)
      xxxx(0,0,0,0)=1.0d+00
      yyyy(0,0,0,0)=1.0d+00
      zzzz(0,0,0,0)=1.0d+00
c
c     loop over shells a b for <aabb>
c
      intshel=nshell+1
      do 100 iashl=intshel,ntshel
      kashl=index2(iashl)
c
c     position of the shell's atom
c
      iaatom=katom(iashl)
      ax=c(1,iaatom)
      ay=c(2,iaatom)
```

```
      az=c(3,iaatom)
c
c     info about the primatives
c
      istarta=kstart(iashl)
      ienda=istarta+kng(iashl)-1
c
c     info about the subshells
c
      iamax=kmax(iashl)
      iamin=kmin(iashl)
      loca =kloc(iashl)-iamin
c
c     find the highest spin value in this shell
c
      katype=ktype(iashl)-1
      jatype=1+iamax-iamin
c
c     b shell
c
      do 101 ibshl=intshel,iashl
c
c     ----- go parallel! -----
c
      if (goparr) then
         ipcount = ipcount + 1
         if (mod(ipcount,nproc).ne.0) go to 101
      end if
      ibatom=katom(ibshl)
      bx=c(1,ibatom)
      by=c(2,ibatom)
      bz=c(3,ibatom)
      istartb=kstart(ibshl)
      iendb=istartb+kng(ibshl)-1
      ibmax=kmax(ibshl)
      ibmin=kmin(ibshl)
      locb = kloc(ibshl)-ibmin
      kbtype=ktype(ibshl)-1
      jbtype=1+ibmax-ibmin
      abrr=(ax-bx)*(ax-bx)+(ay-by)*(ay-by)+(az-bz)*(az-bz)
c
c    end of shells
c
c    loop over guassian primatives while reading
c    in exponents and contractions
c
       call vclr(temps,1,jatype*jbtype*jatype*jbtype)
       do 110 iaprm=istarta,ienda
        a1=ex(iaprm)
        call forcnt(iaprm,iamax,conta)
        do 111 icprm=istarta,ienda
         a2=ex(icprm)
```

```
         call forcnt(icprm,iamax,contc)
         a6=a1+a2
         do 112 ibprm=istartb,iendb
          a3=ex(ibprm)
          call forcnt(ibprm,ibmax,contb)
          if(abrr*a1*a3/(a1+a3) .gt. tol .or.
     *       abrr*a2*a3/(a2+a3) .gt. tol) go to 112
         do 113 ieprm=istartb,iendb
          a5=ex(ieprm)
          call forcnt(ieprm,ibmax,conte)
          a4=a1+a2+a3+a5
          a4inv=1.0d+00/a4
          if(abrr*a1*a5/(a1+a5) .gt. tol .or.
     *       abrr*a2*a5/(a2+a5) .gt. tol) go to 113
          a7=a3+a5
          a10=0.5d+00/a4
          dx=(a1*ax+a2*ax+a3*bx+a5*bx)*a4inv
          dy=(a1*ay+a2*ay+a3*by+a5*by)*a4inv
          dz=(a1*az+a2*az+a3*bz+a5*bz)*a4inv
          dax=dx-ax
          day=dy-ay
          daz=dz-az
          dbx=dx-bx
          dby=dy-by
          dbz=dz-bz
c
c         calculate kabc
c
          abck=((pi*a4inv)*sqrt(pi*a4inv))*exp(-abrr*a6*a7*a4inv)
c
c     generate higher angular momentum xxxx,etc
c
c     recursion based on j
c
      do 2000 j=1,katype
                 xxxx(0,j,0,0)= dax*xxxx(0,j-1,0  ,0  )+a10*(
     *                          (j-1)*xxxx(0,j-2,0  ,0  ))
                 yyyy(0,j,0,0)= day*yyyy(0,j-1,0  ,0  )+a10*(
     *                          (j-1)*yyyy(0,j-2,0  ,0  ))
                 zzzz(0,j,0,0)= daz*zzzz(0,j-1,0  ,0  )+a10*(
     *                          (j-1)*zzzz(0,j-2,0  ,0  ))
 2000 continue
c
c      recursion based on k
c
       do 2010 k=1,kbtype
        do 2010 j=0,katype
                 xxxx(0,j,k,0)= dbx*xxxx(0,j  ,k-1,0  )+a10*(
     *                             j*xxxx(0,j-1,k-1,0  ) +
     *                          (k-1)*xxxx(0,j  ,k-2,0  ))
                 yyyy(0,j,k,0)= dby*yyyy(0,j  ,k-1,0  )+a10*(
     *                             j*yyyy(0,j-1,k-1,0  ) +
```

```
     *                           (k-1)*yyyy(0,j  ,k-2,0  ))
                    zzzz(0,j,k,0)= dbz*zzzz(0,j  ,k-1,0  )+a10*(
     *                               j*zzzz(0,j-1,k-1,0  ) +
     *                           (k-1)*zzzz(0,j  ,k-2,0  ))
 2010  continue
c
c       recursion based on l
c
        do 2020 l=1,kbtype
          do 2020 k=0,kbtype
            do 2020 j=0,katype
                  xxxx(0,j,k,l)= dbx*xxxx(0,j  ,k  ,l-1)+a10*(
     *                               j*xxxx(0,j-1,k  ,l-1) +
     *                               k*xxxx(0,j  ,k-1,l-1) +
     *                           (l-1)*xxxx(0,j  ,k  ,l-2))
                  yyyy(0,j,k,l)= dby*yyyy(0,j  ,k  ,l-1)+a10*(
     *                               j*yyyy(0,j-1,k  ,l-1) +
     *                               k*yyyy(0,j  ,k-1,l-1) +
     *                           (l-1)*yyyy(0,j  ,k  ,l-2))
                  zzzz(0,j,k,l)= dbz*zzzz(0,j  ,k  ,l-1)+a10*(
     *                               j*zzzz(0,j-1,k  ,l-1) +
     *                               k*zzzz(0,j  ,k-1,l-1) +
     *                           (l-1)*zzzz(0,j  ,k  ,l-2))
 2020   continue
c
c         recursion based on i
c
          do 2030 l=0,kbtype
            do 2030 k=0,kbtype
              do 2030 j=0,katype
                do 2030 i=1,katype
                  xxxx(i,j,k,l)= dax*xxxx(i-1,j  ,k  ,l  )+a10*(
     *                           (i-1)*xxxx(i-2,j  ,k  ,l  ) +
     *                               j*xxxx(i-1,j-1,k  ,l  ) +
     *                               k*xxxx(i-1,j  ,k-1,l  ) +
     *                               l*xxxx(i-1,j  ,k  ,l-1))
                  yyyy(i,j,k,l)= day*yyyy(i-1,j  ,k  ,l  )+a10*(
     *                           (i-1)*yyyy(i-2,j  ,k  ,l  ) +
     *                               j*yyyy(i-1,j-1,k  ,l  ) +
     *                               k*yyyy(i-1,j  ,k-1,l  ) +
     *                               l*yyyy(i-1,j  ,k  ,l-1))
                  zzzz(i,j,k,l)= daz*zzzz(i-1,j  ,k  ,l  )+a10*(
     *                           (i-1)*zzzz(i-2,j  ,k  ,l  ) +
     *                               j*zzzz(i-1,j-1,k  ,l  ) +
     *                               k*zzzz(i-1,j  ,k-1,l  ) +
     *                               l*zzzz(i-1,j  ,k  ,l-1))
 2030     continue
c
c
c    loop over subshells to get values of the integrals
c
          imove=0
```

```
          do 120 iasub=iamin,iamax
c          the x coordinate in s(num,num,num,num)
           iaposit=loca+iasub
           l1=ldat(iasub)
           m1=mdat(iasub)
           n1=ndat(iasub)
           contra=conta(iasub)*contc(iasub)
c
           do 121 ibsub=ibmin,ibmax
c           the y coordinate in s(num,num,num,num)
            ibposit=locb+ibsub
            if(ibposit .gt. iaposit) go to 120
            l2=ldat(ibsub)
            m2=mdat(ibsub)
            n2=ndat(ibsub)
            contre=contra*contb(ibsub)*conte(ibsub)
c
              imove=imove+1
              temps(imove) = temps(imove)
     *            +abck*contre*xxxx(l1,l1,l2,l2)*
     *               yyyy(m1,m1,m2,m2)*zzzz(n1,n1,n2,n2)
c
  121      continue
  120     continue
c
c
  113    continue
  112    continue
  111   continue
  110  continue
c
c        loop over temporary matrix
c
         dum=0.0d+00
         imove=0
         do 421 iaposit=iamin+loca,iamax+loca
           do 422 ibposit=ibmin+locb,min(ibmax+locb,iaposit)
              imove=imove+1
              if(temps(imove) .gt. dum) dum=temps(imove)
  422      continue
  421    continue
         s(kashl+ibshl)=dum
c
c     end shell loops
c
  101 continue
  100 continue
c
c     ----- sum up partial contributions if parallel -----
c
      if(goparr)call ddi_gsumf(1039,s,ntshel*(ntshel+1)/2)
c
```

```
      call dawrit(idaf,ioda,s,ntshel*(ntshel+1)/2,342,0)
c
      call tsecnd(tim)
      tim = tim-tim0
      if(some) write(iw,9000) tim
      return
c
 9000 format(1x,'(iijj) integral schwarz inequality overhead: t=',f12...
     .2)
      end
```

C.2 Changes to Preexisting GAMESS and NEO Source Files

All of the modifications to each of the preexisting GAMESS and NEO source files that were necessary to accommodate positrons are listed, including a brief description of each change. In order to save space, only the differences between the modified source code and the `7Sep06R4` release of GAMESS are listed. The boxed text in each section below is the output from `diff`, a standard program on any Unix system that prints the differences between two text files.

For example, if the file, `example1.txt`, contains

```
line 1 in both is "Line 1"
line 2 here is "Line 2"
line 3 in both is "Line 3"
```

and a file, `example2.txt`, contains

```
line 1 in both is "Line 1"
line 2a here is "Line 2a"
line 2b here is "Line 2b"
line 3 in both is "Line 3"
line 4 here is "Line 4"
```

the output of the command

```
diff example1.txt example2.txt
```

will be

```
2c2,3
< line 2 here is "Line 2"
---
> line 2a here is "Line 2a"
> line 2b here is "Line 2b"
3a5
> line 4 here is "Line 4"
```

To make file `example1.txt` match file `example2.txt`, one would need to make two different modifications to `example1.txt`. First, one would need to change (c) line 2 of `example1.txt` to match lines 2 and 3 (2,3) of `example2.txt`. Second, one would need to add (a) line 5 of `example2.txt` to `example1.txt` after line 3.

C.2.1 GAMESS.SRC. The small change to the `GAMESS.SRC` file is to allow the user to request the calculation of the electron-positron annihilation rate using the *LAMBDAHF* routine in `NEOPOS.SRC`. This requires the `POSNEO` and `POSPRP` logical variables. The `IUNIQA` array and `MAXNEO` parameter must come along also since they are part of the `NUCPOS` common block.

```
1528c1528
<       LOGICAL GOPARR,DSKWRK,MASWRK
---
>       LOGICAL GOPARR,DSKWRK,MASWRK,POSNEO,POSPRP
1530a1531
>       PARAMETER (MAXNEO=40)
1541a1543
>       COMMON /NUCPOS/ IUNIQA(MAXNEO),POSNEO,POSPRP
1670a1673
>             IF (POSNEO.AND.POSPRP) CALL LAMBDAHF
```

C.2.2 INPUTA.SRC. Other than the new module in `NEOPOS.SRC`, the modifications to the `INPUTA.SRC` file are the most significant of those required for addition of positrons to GAMESS. These changes were necessary to allow positron basis function centers to be located on classical nuclei, something that was not necessary (or desired) when modeling protons with NEO. When positronic basis function centers are generated with the symmetry operators, a check is made to determine if the generated center already exists. This required the introduction of two logical variables, `QMCHKA` and `QMCHKB`, since we want to check whether the generated center *and* the existing center are quantum or classical.

```
278c278
<       LOGICAL ELEMENT,SPRKLE,DSTNCE
---
>       LOGICAL ELEMENT,SPRKLE,DSTNCE,QMCHKA,QMCHKB,POSNEO,POSPRP
284a285
>       PARAMETER (MAXNEO=40)
333a335
>       COMMON /NUCPOS/ IUNIQA(MAXNEO),POSNEO,POSPRP
1726a1729,1731
>       QMCHKA = .FALSE.
>       QMCHKB = .FALSE.
> C
1731a1737,1747
> C
> C            IF THIS IS A NEORUN W/POSITRONS,
> C            NEED TO SET FLAGS (QMCHKA/B) FOR QUANTUM NUCLEI
> C
>          IF((NEORUN.EQ.1) .AND.POSNEO) THEN
>             QMCHKA=.FALSE.
>             DO J = 1,maxneo
>                IF( IUNIQA(J) .EQ. NAT ) QMCHKA=.TRUE.
>             end do
>          END IF
> C
1732a1749,1754
>             IF((NEORUN.EQ.1).AND.POSNEO) THEN
>                QMCHKB=.FALSE.
>                DO J = 1,maxneo
>                   IF( IAT .EQ. IUNIQA(J) )  QMCHKB=.TRUE.
>                end do
>             END IF
```

(continued on next page)

(continued from previous page)

```
1734c1756,1766
<             IF(TEST .LE. 1.0D-12) GO TO 820
---
> C
> C     FOR NEO RUN WITH POSITRONS, SPECIAL CHECK REQUIRED, FOR IT IS
> C     OK TO HAVE THE POSITRON BASIS SET CENTERED ON AN ACTUAL ATOM.
> C
>             IF((NEORUN.EQ.1).AND.POSNEO) THEN
>                IF ((QMCHKA.EQV.QMCHKB).AND.(TEST.LE.1.0D-12)) THEN
>                   GO TO 820
>                END IF
>             ELSE
>                IF(TEST .LE. 1.0D-12) GO TO 820
>             END IF
1813a1846,1854
> C
> C              NEO RUNS WITH POSITRONS INCLUDED
>          IF((NEORUN.EQ.1).AND.POSNEO) THEN
>             QMCHKA=.FALSE.
>             DO I = 1,maxneo
>                IF( IAT .EQ. IUNIQA(I) ) QMCHKA=.TRUE.
>             ENDDO
>          END IF
> C
1831a1873,1880
> C
>                IF((NEORUN.EQ.1).AND.POSNEO) THEN
>                   QMCHKB=.FALSE.
>                   DO J = 1,20
>                      IF( I .EQ. IUNIQA(J) ) QMCHKB=.TRUE.
>                   ENDDO
>                END IF
> C
1833c1882
<                IF(TEST .GT. TM10) GO TO 1010
---
>                IF(TEST.GT.TM10   .OR.   (QMCHKA.NEQV.QMCHKB)) GO TO 1010
```

C.2.3 MTHLIB.SRC. The change to the `MTHLIB.SRC` file is to allow positron basis function centers to be collocated with classical nuclei or other electronic basis function centers.

```
1070a1071
>       PARAMETER (MAXNEO=40)
1072c1073
<       LOGICAL CROWD,GOPARR,DSKWRK,MASWRK
---
>       LOGICAL CROWD,GOPARR,DSKWRK,MASWRK,POSNEO,POSPRP
1080a1083
>       COMMON /NUCPOS/ IUNIQA(MAXNEO),POSNEO,POSPRP
1126c1132,1137
<          IF (MASWRK) WRITE(IW,9100) CLOSE
---
> C           NEO RUNS FOR POSITRONS ARE ALLOWED TO HAVE POSITRON
> C           BASIS FUNCTION CENTERS ON CLASSICAL NUCLEI, SO WE MUST
> C           ALLOW "CLOSE" CENTERS IN THIS CASE.
> C
>          IF(NFG.EQ.0  .AND.  .NOT.POSNEO) THEN
>            IF (MASWRK) WRITE(IW,9100) CLOSE
```

C.2.4 NEO.SRC. The most significant changes to NEO are found in the `NEO.SRC` file. In addition to introducing the `MAXNEO` parameter to accomodate more than 20 positron basis function centers, we add the capability to read in the `IUNIQA` array and `POSNEO` and `POSPRP` logical variables from NEO input. Also, for positronic NEO runs, the mass is set to `ONE` (the mass of the electron in atomic units). Finally, the `ELPOVR` logical variable is added to allow for the calculation of the electron-positron four-overlap integrals when the user requests the calculation of the annihilation rate (`POSPRP==.TRUE.`).

```
94c94
<       LOGICAL DIRNUC,NUFOCK,SYMNUC,USRDEX
---
>       LOGICAL DIRNUC,NUFOCK,SYMNUC,USRDEX,POSNEO,POSPRP
100a101
>       PARAMETER (MAXNEO=40)
110,113c111,114
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
<       COMMON /NUCMOR/ QMNUCM(20)
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
>       COMMON /NUCMOR/ QMNUCM(MAXNEO)
116c117
<       COMMON /NUCMON/ QNUN(20),QNN(20)
---
>       COMMON /NUCMON/ QNUN(MAXNEO),QNN(MAXNEO)
118a120
>       COMMON /NUCPOS/ IUNIQA(MAXNEO),POSNEO,POSPRP
128c130
<       PARAMETER (NNAM=23)
---
>       PARAMETER (NNAM=26)
136,138c138,141
<      *          8HEXNB    ,8HNEMPLV  ,8HNEOCI   /
<       DATA KQNAM /1,201,203,5,5,201,0,1,1,1,201,0,0,1,0,5,5,0,0,3,
<      *            803,1,5/
---
>      *          8HEXNB    ,8HNEMPLV  ,8HNEOCI   ,8HPOSNEO  ,
>      *          8HPOSPRP  ,8HIUNIQA  /
>       DATA KQNAM /1,401,403,5,5,401,0,1,1,1,401,0,0,1,0,5,5,0,0,3,
>      *            803,1,5,0,0,401/
172c175
<       DO I = 1,20
---
>       DO I = 1,MAXNEO
190a194,195
>       POSNEO = .FALSE.
>       POSPRP = .FALSE.
203,204c208,209
<      *     0,
<      *     0,0,0,0,0,    0,0,0,0,0,
---
>      *     POSNEO,POSPRP,IUNIQA,
>      *     0,0,0,        0,0,0,0,0,
```

(continued on next page)

(continued from previous page)

```
299c304,305
<      *                   HSSUPD,DIRNUC,SYMNUC,QMATLN,USRDEX
---
>      *                   HSSUPD,DIRNUC,SYMNUC,QMATLN,USRDEX,POSNEO,
>      *                   POSPRP
343c349
< C     --- CHECK ONLY QUANTUM PROTONS REQUESTED ---
---
> C     --- CHECK ONLY QUANTUM PROTONS OR POSITRONS REQUESTED ---
344a351,352
> C     IF POSNEO==TRUE THEN
> C     1=POSITRON
357c365
< C     --- FOR NOW FORCE PROTONS ONLY ---
---
> C     --- FOR NOW FORCE PROTONS OR POSITRONS ONLY ---
359c367,368
<       DO I=1,NUNIQN+NAUXNB
---
> c      DO I=1,NUNIQN+NAUXNB
>       DO I=1,MAXNEO
362c371,373
<          IF(II.EQ.1) THEN
---
>          IF(II.EQ.1.AND.POSNEO) THEN
>             QNUN(I) = 'PS'
>          ELSE IF (II.EQ.1) THEN
390a402,409
>          IF ((ITYP.EQ.1) .AND.POSNEO) THEN
>                QMNUCM(I) = ONE
>          END IF
>          IF ((ITYP.GT.1) .AND.POSNEO) THEN
>             WRITE(IW,*) 'ASKED FOR IUNIQT > 1 WITH POSITRON'
>             WRITE(IW,*) 'PLEASE FIX'
>             CALL ABRT
>          END IF
436c455,456
<      *  5X,7HQMTOLN=,1P,E8.1,0P,5X,7HUSRDEX=,L8)
---
>      *  5X,7HQMTOLN=,1P,E8.1,0P,5X,7HUSRDEX=,L8/
>      *  5X,7HPOSNEO=,L8,5X,7HPOSPRP=,L8)
```

(continued on next page)

(continued from previous page)

```
447,448c467,468
<       LOGICAL DO1NUC,DONGES
<       LOGICAL INIT,NUCNUC,ELENUC,DIRNUC,NUCTRN
---
>       LOGICAL DO1NUC,DONGES,POSNEO,POSPRP
>       LOGICAL INIT,NUCNUC,ELENUC,DIRNUC,NUCTRN,ELPOVR
451a472,473
>       PARAMETER (MAXNEO=40)
> C
454,456c476,478
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
457a480
>       COMMON /NUCPOS/ IUNIQA(MAXNEO),POSNEO,POSPRP
510a534
>       ELPOVR = .FALSE.
512c536
<      &            NUCNUC,ELENUC)
---
>      &            NUCNUC,ELENUC,ELPOVR)
527a552
>          ELPOVR = .FALSE.
529c554
<      &               NUCNUC,ELENUC)
---
>      &               NUCNUC,ELENUC,ELPOVR)
```

(continued on next page)

(continued from previous page)

```
533a559,573
>
> C     --- ELECTRON-POSITRON OVERLAP INTEGRALS ---
> c
>       IF( POSNEO  .AND.  POSPRP ) THEN
>          SVGPAR = GOPARR
> C
>          MIXED = .TRUE.
>          NUCNUC = .FALSE.
>          ELENUC = .FALSE.
>          ELPOVR = .TRUE.
>          CALL NJANDK(MIXED,INIT,SCFTYP,DIRNUC,NUCTRN,FAEN,FBEN,
>      &            NUCNUC,ELENUC,ELPOVR)
> C
>          GOPARR = SVGPAR
>       END IF
555a596
>       PARAMETER (MAXNEO=40)
563,565c604,606
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
625a667
>       PARAMETER (maxneo=40)
631,633c673,675
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
740a783
>       PARAMETER (maxneo=40)
743,745c786,788
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
```

C.2.5 NEOBAS.SRC. In addition to introducing the `MAXNEO` parameter to accomodate more than 20 positronic basis function centers, a significant addition was made so

that one can read in contracted positron basis functions. With the latter change, positron basis sets are formatted in the same way as electronic basis sets.

```
42a43
>       PARAMETER (MAXNEO=40)
67,69c68,70
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
71c72
<       COMMON /NUCMON/ QNUN(20),QNN(20)
---
>       COMMON /NUCMON/ QNUN(MAXNEO),QNN(MAXNEO)
136a138,139
>
>          QNN(NQN) = QNUN(II)
304c307
<             IF(NQN.GT.20) GO TO 1540
---
>             IF(NQN.GT.MAXNEO) GO TO 1540
1149a1153
>       PARAMETER (MAXNEO=40)
1162c1166
<       COMMON /NUCMON/ QNUN(20),QNN(20)
---
>       COMMON /NUCMON/ QNUN(MAXNEO),QNN(MAXNEO)
1518a1523,1524
>       CHARACTER*1 SHELLTYPE
>       LOGICAL POSNEO, POSPRP
1524a1531
>       PARAMETER (MAXNEO=40)
1529a1537
>       COMMON /NUCPOS/ IUNIQA(MAXNEO),POSNEO,POSPRP
```

(continued on next page)

(continued from previous page)

```
1540a1549,1583
>
>       IBF=3
>       CALL SEQOPN(IBF,'NUCBAS', 'OLD',.TRUE., 'FORMATTED')
> C
> C     ----- POSITRON -----
> C
>       IF (.NOT.POSNEO) GO TO 115
>       READ(IBF,*) MXPASS
>       K=1
>       DO 110 I=1,MXPASS
>          READ(IBF,*) SHELLTYPE, NGBFS(I)
>          JNGBFS=NGBFS(I)
>          IF ((SHELLTYPE.EQ.'S') .OR.( SHELLTYPE.EQ.'S'))
>      *      ITYPES(I) = 1
>          IF ((SHELLTYPE.EQ.'P') .OR.( SHELLTYPE.EQ.'P'))
>      *      ITYPES(I) = 2
>          IF ((SHELLTYPE.EQ.'D') .OR.( SHELLTYPE.EQ.'D'))
>      *      ITYPES(I) = 3
>          DO 105 J=1,JNGBFS
>             IF ((SHELLTYPE.EQ.'S') .OR.( SHELLTYPE.EQ.'S')) THEN
> C            IF (KSHELLTYPE.EQ.1) THEN
>               READ(IBF,*) NGBF, EEX(K), CCS(K)
>             END IF
>             IF ((SHELLTYPE.EQ.'P') .OR.( SHELLTYPE.EQ.'P')) THEN
> C            IF (KSHELLTYPE.EQ.2) THEN
>               READ(IBF,*) NGBF, EEX(K), CCP(K)
>             END IF
>             IF ((SHELLTYPE.EQ.'D') .OR.( SHELLTYPE.EQ.'D')) THEN
> C            IF (KSHELLTYPE.EQ.3) THEN
>               READ(IBF,*) NGBF, EEX(K), CCD(K)
>             END IF
>             K=K+1
>   105    CONTINUE
>   110 CONTINUE
>   115 IF (POSNEO) GO TO 300
1544,1545d1586
<       IBF=3
<       CALL SEQOPN(IBF,'NUCBAS', 'OLD',.TRUE., 'FORMATTED')
1555c1596
<       IGX = 0
---
>   300 IGX = 0
```

C.2.6 NEOCAS.SRC. Again, just a few changes to the `NEOCAS.SRC` file to accomodate more than 20 positron basis function centers.

```
8a9
>       PARAMETER (MAXNEO=40)
19,21c20,22
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
285a287
>       PARAMETER (MAXNEO=40)
300,302c302,304
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
```

C.2.7 NEOFCI.SRC. The `MAXNEO` parameter is introduced into the `NEOFCI.SRC` file to allow for more than 20 positron basis function centers.

```
117a118
>       PARAMETER (MAXNEO=40)
152,154c153,155
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
777a779
>       PARAMETER (MAXNEO=40)
792,794c794,796
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
```

C.2.8 NEOHF.SRC. The `MAXNEO` parameter is introduced into the `NEOHF.SRC` file to allow for more than 20 positron basis function centers.

```
133a134
>       PARAMETER (MAXNEO=40)
148,150c149,151
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
582a584
>       PARAMETER (MAXNEO=40)
600,602c602,604
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
1500a1503
>       PARAMETER (MAXNEO=40)
1513,1515c1516,1518
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
```

C.2.9 NEOINT.SRC. The `MAXNEO` parameter is introduced into the `NEOINT.SRC` file to allow for more than 20 positron basis function centers. Also, the `neodftint` routine is called for calculation of the four-overlap integrals since all of the necessary setup is already taken care of for the other integral packages.

```
17a18
>       PARAMETER (MAXNEO=40)
39,42c40,43
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
<       COMMON /NUCMOR/ QMNUCM(20)
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
>       COMMON /NUCMOR/ QMNUCM(MAXNEO)
624a626
>       PARAMETER (MAXNEO=40)
648,650c650,652
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
736c738
<      *                  NUCNUC,ELENUC)
---
>      *                  NUCNUC,ELENUC,ELPOVR)
742c744
<       LOGICAL INIT,DIRNUC,NUCTRN,MIXED,NUCNUC,ELENUC
---
>       LOGICAL INIT,DIRNUC,NUCTRN,MIXED,NUCNUC,ELENUC,ELPOVR
843c845,849
<       CALL TWOENI(SCFTYP,DIRNUC,NUCTRN,INTTYP,SCHWRZ,NINT,NSCHWZ,
---
>       IF(ELPOVR) THEN
>          IDFTOVRLP=4
>          CALL NEODFTINT(IDFTOVRLP)
>       ELSE
>          CALL TWOENI(SCFTYP,DIRNUC,NUCTRN,INTTYP,SCHWRZ,NINT,NSCHWZ,
846a853
>       END IF
855,858c862,867
<          IF(MIXED) THEN
<             WRITE(IW,*) ' ..... END OF ELECTRON-NUCLEAR INTEGRALS .....'
<          ELSE
<             WRITE(IW,*) ' ..... END OF NUCLEAR-NUCLEAR INTEGRALS .....'
---
>          IF(MIXED.AND.ELENUC.AND..NOT.ELPOVR) THEN
>            WRITE(IW,*)' .. END OF ELECTRON-NUCLEAR COULOMB INTEGRALS ..'
>          ELSE IF(MIXED.AND..NOT.ELENUC.AND.ELPOVR) THEN
>          WRITE(IW,*)' '
>          ELSE
>            WRITE(IW,*)' ..... END OF NUCLEAR-NUCLEAR INTEGRALS ......'
```

C.2.10 NEOMP2.SRC. The MAXNEO parameter is introduced into the NEOMP2.SRC file to allow for more than 20 positron basis function centers. Also, a dummy variable is added to the NJANDK call to accomodate the calculation of the four-overlap integrals.

```
51c51
<      *                  .FALSE.,.TRUE.)
---
>      *                  .FALSE.,.TRUE.,.FALSE.)
72a73
>       PARAMETER (MAXNEO=40)
87,89c88,90
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
```

C.2.11 NEOSTB.SRC. This is just the 'stub' file to abort NEO runs when the binary doesn't have NEO compiled into it. In addition to the changes to the NEO input deck, there is now a stub for the LAMBDAHF routine for computing the annihilation rate.

```
25c25
<       LOGICAL NEOOPT,NEOHSS
---
>       LOGICAL NEOOPT,NEOHSS,POSNEO,POSPRP
29a30
>       PARAMETER (MAXNEO=40)
38,41c39,42
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
<       COMMON /NUCMOR/ QMNUCM(20)
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
>       COMMON /NUCMOR/ QMNUCM(MAXNEO)
44d44
<       COMMON /NUCOPT/ NEOOPT,NUSTEP,NEOTS
```

(continued on next page)

(continued from previous page)

```
45a46,47
>       COMMON /NUCOPT/ NEOOPT,NUSTEP,NEOTS
>       COMMON /NUCPOS/ IUNIQA(MAXNEO),POSNEO,POSPRP
55c57
<       PARAMETER (NNAM=22)
---
>       PARAMETER (NNAM=25)
63,65c65,69
<      *          8HEXNB    ,8HNEMPLV  /
<       DATA KQNAM /1,201,203,5,5,201,0,1,1,1,201,0,0,1,0,5,5,0,0,3,
<      *            803,1/
---
>      *          8HEXNB    ,8HNEMPLV  ,8HPOSNEO  ,8HPOSPRP  ,
>      *          8HIUNIQA  /
>       DATA KQNAM /1,201,203,5,    5,201,0,1,    1,1,201,0,
>      *            0,1,0,5,        5,0,0,3,      803,1,0,0,
>      *            201/
101a106
>          IUNIQA(I) = 0
117a123,124
>       POSNEO = .FALSE.
>       POSPRP = .FALSE.
125,127c132,134
<      *     HSSUPD,DIRNUC,SYMNUC,QMTOLN,EXNB,NEMPLV,
<      *     0,0,
<      *     0,0,0,0,0,    0,0,0,0,0,
---
>      *     HSSUPD,DIRNUC,SYMNUC,QMTOLN,EXNB,NEMPLV,POSNEO,POSPRP,
>      *     IUNIQA,
>      *       0,0,0,0,    0,0,0,0,0,
260a268,273
> C*MODULE NEOPOS   *DECK LAMDAHF
>       SUBROUTINE LAMBDAHF
>       IMPLICIT DOUBLE PRECISION(A-H,O-Z)
>       CALL ABRT
>       RETURN
>       END
```

C.2.12 NEOSYM.SRC. The `MAXNEO` parameter is introduced into the `NEOSYM.SRC` file to allow for more than 20 positron basis function centers.

```
14a15
>       PARAMETER (MAXNEO=40)
24,26c25,27
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
113a115
>       PARAMETER (MAXNEO=40)
133,135c135,137
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
653a656
>       PARAMETER (MAXNEO=40)
666,668c669,671
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
889a893
>       PARAMETER (MAXNEO=40)
902,904c906,908
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
1002a1007
>       PARAMETER (MAXNEO=40)
```

(continued on next page)

(continued from previous page)

```
1017,1019c1022,1024
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
1668a1674,1675
> c
>       PARAMETER (MAXNEO=40)
1672,1674c1679,1681
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
```

C.2.13 NEOTRN.SRC. The `MAXNEO` parameter is introduced into the `NEOTRN.SRC` file to allow for more than 20 positron basis function centers.

```
11a12
>       PARAMETER (MAXNEO=40)
39,41c40,42
<       COMMON /NUCMOI/ NUNIQN,IUNIQN(20),IUNIQT(20),NQMNUC,IQMNUC(20),
<      *                IQNTYP(20),NUMNB,NUCST,NAUXNB,IAUXNB(20),NUMULT,
<      *                NNA,NNB,NTAUXB
---
>       COMMON /NUCMOI/ NUNIQN,IUNIQN(MAXNEO),IUNIQT(MAXNEO),NQMNUC,
>      *                IQMNUC(MAXNEO),IQNTYP(MAXNEO),NUMNB,NUCST,NAUXNB,
>      *                IAUXNB(MAXNEO),NUMULT,NNA,NNB,NTAUXB
```

Bibliography

1. Barnes, L. D. *Energy-resolved positron-molecule annihilation: Vibrational Feshbach resonances and bound states*. Dissertation, University of California, San Diego, 2005.

2. Barnes, L. D., J. A. Young, and C. M. Surko. "Energy-resolved positron annihilation rates for molecules". *Phys. Rev. A*, 74(1):012706, Jul 2006.

3. Behringer, R. and C. G. Montgomery. "The Angular Distribution of Positron Annihilation Radiation". *Phys. Rev.*, 61:222–224, 1942.

4. Beling, C.D., S. Fung, Li Ming, M. Gong, and B.K. Panda. "A theoretical search for possible high efficiency semiconductor based field assisted positron moderators". *Applied Surface Science*, 149:253–259, 1999.

5. Bernath, P. F. *Spectra of Atoms and Molecules*, chapter Vibrational Spectroscopy, 206. Oxford University Press, 1995.

6. Bečvář, F. Private Communication.

7. Bode, B. M. and M. S. Gordon. "Macmolplt: a graphical user interface for GAMESS". *J. Mol. Graphics Mod.*, 16:133–138, 1998.

8. Buenker, R.J., H.-P. Liebermann, V. Melnikov, M. Tachikawa, L. Pichl, and M. Kimura. "Positron Binding Energies for Alkali Hydrides". *J. Phys. Chem. A*, 109(26):5956–5964, 2005.

9. "Computational Chemistry Comparison and Benchmark DataBase". URL `http://srdata.nist.gov/cccbdb`.

10. Cook, D. B. *Handbook of Computational Quantum Chemistry*. Dover, 2005.

11. DeBenedetti, S., C. E. Cowan, W. R. Konneker, and H. Primakoff. "On the Angular Distribution of Two-Photon Annihilation Radiation". *Phys. Rev.*, 77(2):205–212, 1950.

12. Deutsch, M. "Evidence for the formation of positronium in gases". *Phys. Rev.*, 82:455–456, 1951.

13. Deutsch, M. "Three-quantum decay of positronium". *Phys. Rev.*, 83:866–867, 1951.

14. Dirac, P. A. M. "The Quantum Theory of the Electron". *Proc. Roy. Soc.*, 117:610, 1928.

15. Dunning, T. H., Jr. "Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen". *J. Chem. Phys.*, 90(2):1007–1023, 1989.

16. Emin, D. "Unusual properties of icosahedral boron-rich solids". *J. Sol. State. Chem.*, 179:2791–2798, 2006.

17. Ferrell, R. A. "Thoery of Positron Annihilation in Solids". *Rev. Mod. Phys.*, 28(3):308–337, 1956.

18. Feshbach, H. "Unified theory of nuclear reactions". *Ann. of Phys.*, 5:357–390, 1958.

19. Goldanskii, V. I. and Y. S. Sayasov. "On the resonance annihilation of positrons in collisions with neutral atoms or molecules". *Phys. Lett.*, 13:300–301, December 1964.

20. Green, T.A., A.C. Switendick, and David Emin. "*Ab initio* self-consistent field (SCF) calculations on borane icosahedra with zero, one, or two substituted carbon atoms". *J. Chem. Phys.*, 89(11):6815–6822, 1988.

21. Gribakin, G. F. "Mechanisms of positron annihilation on molecules". *Phys. Rev. A*, 61(2):022720, Jan 2000.

22. Gribakin, G. F. and C. M. R. Lee. "Positron Annihilation in Molecules by Capture into Vibrational Feshbach Resonances of Infrared-Active Modes". *Phys. Rev. Lett.*, 97(19):193201, 2006.

23. Hunt, A. W., D. B. Cassidy, P. A. Sterne, T. E. Cowan, R. H. Howell, K. G. Lynn, and J. A. Golovchenko. "Doppler Broadening of In-Flight Positron Annihilation Radiation dueo to Electron Momentum". *Phys. Rev. Let.*, 86(24):5612–5615, 2001.

24. Iordanov, T. and S. Hammes-Schiffer. "Vibrational analysis for the nuclear–electronic orbital method". *J. Chem. Phys.*, 118:9489–9496, June 2003.

25. Jean, Y. C., P. E. Mallon, and D. M. Schrader. *Principles and Applications of Positron and Positronium Chemistry*, chapter Introduction to Positron and Positronium Chemistry. World Scientific, 2003.

26. Kirkegaard, P., M. Eldrup, O. E. Mogensen, and N. J. Pedersen. "Program system for analysing positron lifetime spectra and angular correlation curves". *Computer Physics Communications*, 23:307–335, July 1981.

27. Kirkegaard, P., N. Pederson, and M. Eldrup. *PATFIT-88*. Technical Report Riso-M-2704, Risø National Laboratory, 1989.

28. Knoll, G. F. *Radiation Detection and Measurement*. John Wiley and Sons, Inc., New York, 2000.

29. Knuth, D. E. and S. Levy. *The* `CWEB` *System of Structured Documentation*. Addison-Wesley, Reading, Massachusetts, 1993.

30. Kozlova, S. G., N. K. Moroz, and V. V. Volkov. "Structural features of potassium and cesium dodecahydro-closo-dodecaborates(2-) according to ^{1}H and ^{11}B NMR data". *J. Struc. Chem.*, 34:244–247, March 1993.

31. Krause-Rehberg, R. and H. S. Leipner. *Positron Annihilation in Semiconductors: Defect Studies*, volume 12 of *Solid-State Physics*. Springer, New York, 1999.

32. Krishnan, R., J. S. Binkley, R. Seeger, and J. A. Pople. "Self-consistent molecular orbital methods. XX. A basis set for correlated wave functions". *J. Chem. Phys.*, 72(1):650–654, 1980.

33. Kurtz, H. A. and K. D. Jordan. "Theoretical studies of positron–molecule complexes". *J. Chem. Phys.*, 75(4):1876–1887, 1981.

34. Lee, C. "Stationary states of electron-positron systems and annihilation transitions". *Sov. Phys. JETP*, 33(2):281–291, 1958.

35. Mella, M., D. Bressanini, and G. Morosi. "Stability and production of positron-diatomic molecule complexes". *J. Chem. Phys.*, 114(24):10579–10582, 2001.

36. Mijnarends, P. E. and A. Bansil. *Proceedings of the International School of Physics (Enrico Fermi): Positron Spectroscopy of Solids*, chapter Theory of Electron and Positron Momentum Distributions in Solids. IOS Press, Amsterdam, 1995.

37. Mitroy, J. "Energy and expectation values of the PsH system". *Phys. Rev. A*, 73(5):054502, 2006.

38. Mitroy, J., M. W. J. Bromley, and G. G. Ryzhikh. "Positron and positronium binding to atoms". *J. Phys. B.*, 35:R81–R116, 2002.

39. Mitroy, J and G G Ryzhikh. "The structure of e $^+$ LiH". *J. Phys. B: At. Mol. Opt.*, 33(18):3495–3506, 2000.

40. Møller, C. and M. S. Plesset. "Note on an Approximation Treatment for Many-Electron Systems". *Phys. Rev.*, 46(7):618–622, Oct 1934.

41. Muller, R. "PyQuante©: Python Quantum Chemistry". URL `http://pyquante.sourceforge.net`.

42. Olsen, J. V., P. Kirkegaard, N. J. Pedersen, and M. Eldrup. *PALSfit: A computer program for analysing positron lifetime spectra*. Technical Report Risø-R-0000(EN), Risø National Laboratory, 2006.

43. Pak, M. V. and S. Hammes-Schiffer. "Electron-Proton Correlation for Hydrogen Tunneling Systems". *Phys. Rev. Lett.*, 92:103002, 2004.

44. Pak, M. V., C. Swalina, S. P. Webb, and S. Hammes-Schiffer. "Application of the nuclear-electronic orbital method to hydrogen transfer systems: multiple centers and multiconfigurational wavefunctions". *Chem. Phys.*, 304:227–236, 2004.

45. Paul, D. A. L. and L. Saint-Pierre. "Rapid Annihilations of Positrons in Polyatomic Gases". *Phys. Rev. Lett.*, 11(11):493–496, Dec 1963.

46. Reyes, A., M. V. Pak, and S. Hammes-Schiffer. "Investigation of isotope effects with the nuclear-electronic orbital approach". *J. Chem. Phys.*, 123:064104, 2005.

47. Saito, H., Y. Nagashima, T. Kurihara, and T. Hyodo. "A new positron lifetime spectrometer using a fast digital oscilloscope and BaF_2 scintillators". *Nucl. Instr. and Meth. A*, 487:612–617, 2002.

48. Sakurai, J. J. *Advanced Quantum Mechanics*. Addison-Wesley Publishing Company, 1967.

49. Schmidt, M. W., K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, M. Dupuis S. J. Su, T. L. Windus, and J. A. Montgomery. "GAMESS Version 19 May 2004 (R4)". *J. Comput. Chem.*, 14:1347–1363, 1993.

50. Schmidt, M. W. and K. Ruedenberg. "Effective convergence to complete orbital bases and to the atomic Hartree–Fock limit through systematic sequences of Gaussian primitives". *J. Chem. Phys.*, 71(10):3951–3962, 1979.

51. Schrader, D. M. "Self-Consistent-Field Theory for One-Positron Many-Electron Systems". *Phys. Rev. A*, 1(4):1070–1080, Apr 1970.

52. Schrader, D. M. *Principles and Applications of Positron and Positronium Chemistry*, chapter Compounds of Positrons and Positronium. World Scientific, 2003.

53. Shearer, J. W. and M. Deutsch. "The lifetime of positronium in matter". *Phys. Rev.*, 76:462, 1949.

54. Shoemaker, J.R., L.W. Burggraf, and M.S. Gordon. "An Integrated Molecular Orbital/-Molecular Mechanics Optimization Scheme For Surfaces, SIMOMM". *J. Phys. Chem. A*, 103:3245, 1999.

55. Smith, P. M. and D. A. L. Paul. "Positron annihilation in methane gas". *Can. J. of Phys.*, 48:2984–2990, 1970.

56. Sterne, P. A., J. E. Pask, and B. M. Klein. "Calculation of positron observables using finite element-based approach". *App. Surf. Sci.*, 149:238–243, 1999.

57. Stoll, H., P. Castellaz, and A. Siegle. *Principles and Applications of Positron and Positronium Chemistry*, chapter AMOC in Positron and Positronium Chemistry. World Scientific, 2003.

58. Strasburger, K. "Adiabatic positron affinity of LiH". *J. Chem. Phys.*, 114(2):615–616, 2001.

59. Swalina, C. and S. Hammes-Schiffer. "Impact of Nuclear Quantum Effects on the Molecular Structure of Bihalides and the Hydrogen Fluoride Dimer". *J. Phys. Chem. A*, 109:10410, 2005.

60. Swalina, C., M. V. Pak, and S. Hammes-Schiffer. "Alternative formulation of many-body perturbation theory for electron-proton correlation". *Chem. Phys. Lett.*, 404:394–399, 2005.

61. Swalina, C., M. V. Pak, and S. Hammes-Schiffer. "Analysis of the nuclear-electronic orbital method for model hydrogen transfer systems". *J. Chem. Phys.*, 123:014303, 2005.

62. Swalina, C., M.V. Pak, A. Chakraborty, and S. Hammes-Schiffer. "Explicit Dynamical Electron-Proton Correlation in the Nuclear-Electronic Orbital Framework". *J. Phys. Chem. A*, 110(33):9983–9987, 2006.

63. Szabo, A. and N. S. Ostlund. *Modern Quantum Chemistry*. McGraw-Hill, Inc., London, 1989.

64. Tachikawa, M. "Simultaneous optimization of Gaussian type function exponents for electron and positron with full-CI wavefunction - application to ground and excited states of positronic compounds with multi-component molecular orbital approach". *Chem. Phys. Lett.*, 350:269–276, December 2001.

65. Takatsuka, A. and S. Ten-no. "Theoretical Study of Positronium Atoms Using Frozen Gaussian-type Geminals". *Korean Chem. Soc.*, 24(6):859–863, 2003.

66. Taketa, H., S. Huzinaga, and K. O-ohata. "Gaussian-Expansion Methods for Molecular Integrals". *J. Phys. Soc. Jap.*, 21(11):2313–2324, 1966.

67. Tiritiris, I. *Investigations on reactivity, structure and dynamic of ionic closo-dodecaborates*. Dissertation, Institut für Anorganische Chemie, 2003.

68. Urban-Klaehn, J., S. Ritchie, and A. W. Hunt. "Application of the surface and volumetric positron techniques for the assessment of material damage in real life conditions". *XIV International Conference on Positron Annihilation*. 2006.

69. Wales, D. J. "Rearrangement mechanisms of B12H122- and C2B10H12". *J. Am. Chem. Soc.*, 115(4):1557–1567, 1993.

70. Weaver, H. Joseph. *Applications of Discrete and Continuous Fourier Analysis*. John Wiley and Sons, 1983.

71. Webb, S. P., T. Iordanov, and S. Hammes-Schiffer. "Multiconfigurational nuclear-electronic orbital approach: Incorporation of nuclear quantum effects in electronic tructure calculations". *J. Chem. Phys.*, 117:4106–4118, September 2002.

72. Weber, M. H., K. G. Lynn, B. Barbiellini, P. A. Sterne, and A. B. Denison. "Direct observation of energy-gap scaling law in CdSe quantum dots with positrons". *Phys. Rev. B*, 66(041305(R)), 2002.

73. Wetter, M. *GenOpt® Generic Optimization Program, Version 2.0*. Technical Report LBNL-54199, Lawrence Berkely National Laboratory, 2004.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
XX–06–2007	Doctoral Dissertation	Oct 2003 — Jun 2007

4. TITLE AND SUBTITLE

A General Quantum Mechanical Method to Predict Positron Spectroscopy

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Adamson, Paul, E., Captain, USAF

5d. PROJECT NUMBER

F1ATA06096J001

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/DS/ENP/07-04

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR/NL
ATTN: Dr. Michael R. Berman
801 N. Randolph St., Bldg 732
Arlington, VA 22203-1977
(703) 696-7781

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The NEO method was modified and extended to positron systems. NEO-MP2 energies and annihilation rates were calculated for PsH, and the effects of basis set size on correlation energies captured with the NEO-MP2 and NEO-FCI methods are compared and discussed. Equilibrium geometries and vibrational energy levels were computed for the LiX and e^+LiX (X = H, F, Cl) systems at the MP2 and NEO-MP2 levels. It was found that anharmonicity plays a significant role, specifically in the differences between the vibrational energy levels of the LiX and e^+LiX systems. The implications of these results with respect to VFR for these systems is discussed. The positron lifetime in $K_2B_{12}H_{12}\cdot CH_3OH$ was measured to be 0.2645 ± 0.0077 ns. Quantum mechanical calculations reveal a spherically symmetric positronic wavefunction, with a peak in the positron density at the outside edge of the hydrogen atom cage. The experimentally determined annihilation rate corresponds to an effective number of electrons of 1.88, indicating that there is significant positron density both inside and outside of the $B_{12}H_{12}$ dianon cage.

15. SUBJECT TERMS

positrons, antiparticles, annihilation radiation, boranes, molecular vibration

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Larry W. Burggraf, AFIT/ENP
U	U	U	UU	206	19b. TELEPHONE NUMBER *(include area code)* (937) 255–3636 x4503